Lecture-15

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

Homework (Due March 25)

• 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 $\lambda_1, \ldots, \lambda_n$ we have

$$\| x_{k+1} - x^* \|_A^2 \leq \left( \frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \| x_0 - x^* \|_A^2$$
Proof

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

$$\tau_1 < \tau_2, \ldots, < \tau_k$$

and $\tau_{k+1} = \frac{\lambda_{n-k} + \lambda_1}{2}$

Define polynomial:

$$Q_{k+1}(\lambda) = \frac{(-1)^{k+1}}{\tau_1 \tau_2 \ldots \tau_k} (\lambda - \tau_1)(\lambda - \tau_2) \ldots (\lambda - \tau_k)(\lambda - \tau_{k+1})$$

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

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

Is polynomial of degree $k+1$ with root at

$$P_k = \frac{(Q_{k+1}(\lambda) - 1)}{\lambda}$$

Degree $k$

$$\min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \quad \text{(B)}$$

$$0 \leq \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \leq \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 = \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2$$

Homework: show this

Example

Clustered around 1

$\{\lambda_1, \ldots, \lambda_{n-m}\}, \{\lambda_{n-m+1}, \ldots, \lambda_n\}$

$m$ largest eigenvalues

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

For small value of $c$

CG will converge in only $m+1$ steps.

$$\| x_{k+1} - x^* \|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \| x_0 - x^* \|_A^2$$
Example

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

\[ N=14, \text{ has four clusters of eigenvalues: single eigenvalues at } 140, 120, \text{ a cluster of 10 eigenvalues very close to 10 with the remaining eigenvalues clustered between .95 and 1.05.} \]
Convergence using Condition number

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

\[ \kappa(A) = \| A \|_2 \| A^{-1} \|_2 = \frac{\lambda_1}{\lambda_n} \]

\[ \lambda_1 > \lambda_n \]

---

Convergence Rate of Steepest Descent: Quadratic Function

\[ \| x_{k+1} - x^* \|_Q^2 \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \| x_k - x^* \|_Q^2 \]

Theorem 3.3

\[ \lambda_1 < \lambda_n \]
What is desirable?

- Matrix $A$ should have either:
  - Few distinct eigenvalues
  - Few distinct eigenvalues, and few clusters of eigenvalues
  - Smaller condition number

Preconditioning

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

Original problem:
\[ \phi(x) = \frac{1}{2} x^T Ax - b^T x \quad \text{or} \quad Ax = b \]

Transformation:
\[ \hat{x} = Cx \quad C^{-1} \hat{x} = x \]

Transformed problem:
\[ \hat{\phi}(\hat{x}) = \frac{1}{2} (C^{-1} \hat{x})^T AC^{-1} \hat{x} - b^T C^{-1} \hat{x} \]
\[ \hat{\phi}(\hat{x}) = \frac{1}{2} \hat{x}^T (C^{-T} AC^{-1}) \hat{x} - (C^{-T} b)^T \hat{x} \quad (C^{-T} AC^{-1}) \hat{x} = (C^{-T} b) \]

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
\[ C^{-T} AC^{-1} = L^{-1} AL^{-T} = L^{-1} LL^T L^{-T} = I \]
Algorithm 5.3 (Preconditioned CG)

Given $x_0$, preconditioner $M$;
set $r_0 \leftarrow Ax_0 - b$;
solve $M_0^t = r_0$ for $y_0$;
$p_0 \leftarrow -r_0, k \leftarrow 0$

While $r_k \neq 0$

\[ M = C^T C \]
\[
\alpha_k \leftarrow -\frac{r_k^T y_k}{p_k^T A p_k};
\]
\[
x_{k+1} \leftarrow x_k + \alpha_k p_k;
\]
\[
r_{k+1} \leftarrow r_k + \alpha_k A p_k;
\]
\[
M y_{k+1} = r_{k+1};
\]
\[
\beta_k \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k};
\]
\[
p_{k+1} \leftarrow y_{k+1} + \beta_k p_k;
\]
\[
k \leftarrow k + 1;
\]
end(while)

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 $\beta$
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 $\alpha_k$;

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

evaluate $\nabla f_{k+1}$;

$\beta_{k+1}^{FR} \leftarrow \frac{\nabla f_{k+1}^2}{\nabla f_k^2}$;

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

$k \leftarrow k + 1$

end(while)

Given $x_0$;

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

While $r_k \neq 0$

$\alpha_k \leftarrow -\frac{\nabla f_k^2}{Ap_k}$;

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

$r_{k+1} \leftarrow r_k + \alpha_k Ap_k$

$\beta_{k+1} \leftarrow \frac{r_{k+1}^2}{r_k^2}$;

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

$k \leftarrow k + 1$

end(while)

5.4

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

\[ p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k \]

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

\[
\begin{align*}
p_k &= -\nabla f_k + \beta_k^{FR} p_{k-1} \\
\nabla^T f_k p_k &= -\nabla^T f_k \nabla f_k + \beta_k^{FR} \nabla^T f_k p_{k-1} \\
\nabla^T f_k p_k &= -\|\nabla f_k\|^2 + \beta_k^{FR} \nabla^T f_k p_{k-1}
\end{align*}
\]

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

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

Choice of step length

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

\[
f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla^T f_k p_k, \quad c_1 \in (0, 1)
\]

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

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

\( \nabla^T f_k p_k < 0 \)
Polak-Ribiere

\[ \beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k} \]

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

\[ \beta_{k+1}^{HS} = \max(\beta_{k+1}^{PR}, 0) \]

This can satisfy descent property

\[ \beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^T p_k} \]

Yet another choice
Quadratic Termination & Restarts

Non-linear CG methods preserve 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:

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

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

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

$$\frac{\|\nabla f_k^T \nabla f_{k+1}\|}{\|\nabla f_k\|} \geq \nu, \quad \nu = .1$$

Theorem 5.3

Two consecutive gradients are far from orthogonal.

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

Another restarting strategy

Results

Termination conditions:
Or 10,000 iterations

<table>
<thead>
<tr>
<th>Problem</th>
<th>$n$</th>
<th>Alg FR IT/f</th>
<th>Alg PR IT/f</th>
<th>Alg PR+ IT/f</th>
<th>mod</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALCVAR3</td>
<td>200</td>
<td>280/5617</td>
<td>2631/5263</td>
<td>2631/5263</td>
<td>0</td>
</tr>
<tr>
<td>GENROS</td>
<td>500</td>
<td>*</td>
<td>1068/2151</td>
<td>1067/2149</td>
<td>1</td>
</tr>
<tr>
<td>XPOWSING</td>
<td>1000</td>
<td>533/1102</td>
<td>212/473</td>
<td>97/229</td>
<td>3</td>
</tr>
<tr>
<td>TRIDIA1</td>
<td>1000</td>
<td>264/551</td>
<td>262/527</td>
<td>262/527</td>
<td>0</td>
</tr>
<tr>
<td>MSQRT1</td>
<td>1000</td>
<td>422/649</td>
<td>113/231</td>
<td>113/231</td>
<td>0</td>
</tr>
<tr>
<td>XPOWELL</td>
<td>1000</td>
<td>568/1175</td>
<td>212/473</td>
<td>97/229</td>
<td>3</td>
</tr>
<tr>
<td>TRIGON</td>
<td>1000</td>
<td>231/467</td>
<td>40/92</td>
<td>40/92</td>
<td>0</td>
</tr>
</tbody>
</table>

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

Given $x_i$;

- evaluate $f_k = f(x_k), \nabla f_k = \nabla f(x_k)$
- set $p_i \leftarrow -\nabla f(x_k), k \leftarrow 0$

While $\nabla f_k \neq 0$

- compute $\alpha_i$;
- $x_{i+1} \leftarrow x_i + \alpha_i p_i$;
- evaluate $\nabla f_{i+1}$;

$$\beta_{i+1}^{PR} \leftarrow \frac{\nabla f_{i+1}^T \nabla f_{i+1}}{\nabla f_i^T \nabla f_i}$$

/*

$$\beta_{i+1}^{PR} \leftarrow \frac{\nabla f_{i+1} (\nabla f_{i+1} - \nabla f_i) \_i}{\nabla f_i^T \nabla f_i}$$

$p_{i+1} \leftarrow -\nabla f_{i+1} + \beta_{i+1}^{PR} p_i$;

$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 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.
Proof

Eigenvalues
\( \lambda_1, \ldots, \lambda_{n-k}, \lambda_{n-k+1}, \ldots, \lambda_n \)

Select polynomial of degree \( k \) such that
\( Q \) has roots at \( k \) largest eigenvalues
\( \lambda_n, \lambda_{n-1}, \ldots, \lambda_{n-k+1} \)
As well as at mid point \( \lambda_1 \) and \( \lambda_{n-k} \)

\[ Q_{k+1}(\lambda) = 1 + \lambda P_k(\lambda) \]

Maximum value attained by \( Q \) on the remaining eigenvalues is precisely

\[ \left( \frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right) \]

\( (C) \)

\[ \| x_{k+1} - x^* \|_d^2 \leq \min_{1 \leq i \leq k} \max \left[ 1 + \lambda_i P_k(\lambda_i) \right] \| x_0 - x^* \|_d^2 \]

\[ \| x_{k+1} - x^* \|_d^2 \leq \left( \frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \| x_0 - x^* \|_d^2 \]

Homework: show this