Numerical Optimization
Lecture 13: Conjugate Gradient Method

Sangkyun Lee

The content is from Nocedal and Wright (2006). Topics marked with ** are optional.

1 Generation of Conjugate Directions

We discuss how to generate a set of a direction \( p_k \) conjugate to \( \{p_0, p_1, \ldots, p_{k-1}\} \), but using the information of \( p_{k-1} \) and not the other ones.

In the CG method, we choose

\[
p_k = -r_k + \beta_k p_{k-1},
\]

where the scalar \( \beta_k \) is chosen so that \( p_k \) and \( p_{k-1} \) must be conjugate w.r.t. \( A \), i.e., \( p_k^T A p_{k-1} = 0 \). From this we have

\[
p_{k-1}^T A ( -r_k + \beta_k p_{k-1} ) = 0,
\]

\[
\Rightarrow \beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}.
\]

For the first direction \( p_0 \), we choose \( p_0 = -\nabla \phi(x_0) = -r_0 \). A preliminary version of the CG algorithm is shown in Algorithm 1. Approximate numbers of FLOPs are shown in the parentheses, assuming that \( A p_k \) and \( p_k^T A p_k \) are cached and multiply-accumulation operation \( \text{MAC}(a, b, c) : a \leftarrow a + (b \ast c) \) is a single operation. The total no. of FLOPs of this algorithm will be approximately \( 2n^2 + 5n \).

2 Conjugacy and the Krylov Subspace

We still need to show that the directions \( p_0, p_1, \ldots, p_{n-1} \) generated by Algorithms 1 and (2) are conjugate wrt \( A \). If so, then by Theorem 11.3, this algorithm will terminate in \( n \) steps. The next theorem shows this property, along with two other important properties: (i) the residuals \( r_i \) are mutually orthogonal, and (ii) each \( p_k \) and \( r_k \) is contained in the Krylov subspace of degree \( k \) for \( r_0 \), defined by

\[
K(r_0; k) = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}.
\]

To understand its relation to CG, we can see from the fact that \( r_0 = Ax_0 - b \) and \( A^{-1} \) can be written due to the Cayley-Hamilton theorem,

\[
A^{-1} \approx \frac{(-1)^{k-1}}{\det(A)} (A^{k-1} + d_{k-2}A^{k-2} + \cdots + d_0I).
\]
Algorithm 1: CG: Preliminary Version

1 Input: \( x_0, A, b \);
2 \( r_0 = Ax_0 - b, p_0 = -r_0, k = 0; \)
3 while \( r_k \neq 0 \) do
4 \( \alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} \) (\( n^2 + 2n \) FLOPs);
5 \( x_{k+1} = x_k + \alpha_k p_k \) (\( n \) FLOPs);
6 \( r_{k+1} = Ar_{k+1} - b \) (\( n^2 \) FLOPs);
7 \( \beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} \) (\( n \) FLOPs);
8 \( p_{k+1} = -r_{k+1} + \beta_{k+1} p_k \) (\( n \) FLOPs);
9 \( k = k + 1; \)
10 end
11 Output: \( x^* = x_k. \)

Therefore,
\[
A^{-1}r_0 \approx \delta(A^{k-1} + d_{k-2}A^{k-2} + \cdots + d_0 I)r_0
\]
where
\[
A^{-1}r_0 = A^{-1}(Ax_0 - b) = x_0 - A^{-1}b \approx x_0 - x_k = c_0p_0 + \cdots + c_{k-1}p_{k-1}.
\]
This gives a rough idea of the definition of the Krylov subspace and why (13.3) below would hold.

**Theorem 13.1.** Suppose that \( x_k \) from the CG algorithm is not the solution \( x^* \). Then the following properties hold:

\[
\begin{align*}
& r_k^T r_i = 0, \quad i = 0, 1, \ldots, k - 1, \quad (13.1) \\
& \text{span}\{r_0, r_1, \ldots, r_k\} = \mathcal{K}(r_0; k), \quad (13.2) \\
& \text{span}\{p_0, p_1, \ldots, p_k\} = \mathcal{K}(r_0; k), \quad (13.3) \\
& p_k^T A p_i = 0, \quad i = 0, 1, \ldots, k - 1. \quad (13.4)
\end{align*}
\]

Therefore, \( \{x_k\} \) converges to \( x^* \) in at most \( n \) steps.

**Proof.** The last three statements: proof by induction. At \( k = 1 \), all three statements are true. Suppose that they are true for some \( k \) (induction hypothesis). We show these three statements hold for \( k + 1 \).

From the induction hypothesis, we have
\[
r_k \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}, \quad p_k \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}.
\]
Then
\[
A p_k \in \text{span}\{Ar_0, A^2 r_0, \ldots, A^{k+1} r_0\}.
\]
Using \( r_{k+1} = r_k + \alpha_k A p_k \), we see that
\[
r_{k+1} \in \text{span}\{r_0, Ar_0, \ldots, A^{k+1} r_0\}.
\]
Combining this with the induction hypothesis, we conclude that
\[
\text{span}\{r_0, r_1, \ldots, r_k, r_{k+1}\} \subseteq \text{span}\{r_0, Ar_0, \ldots, A^{k+1} r_0\}.
\]
To show the reverse inclusion, we start from the induction hypothesis which gives
\[
A^k r_0 \in \text{span}\{p_0, p_1, \ldots, p_k\}
\]
and therefore
\[ A^{k+1}r_0 \in \text{span}\{Ap_0, Ap_1, \ldots, Ap_k\}. \]
From \( r_{i+1} = r_i + \alpha_i Ap_i \), we have \( Ap_i = (r_{i+1} - r_i)/\alpha_i \) for \( i = 0, 1, \ldots, k \), and thus
\[ A^{k+1}r_0 \in \text{span}\{r_0, r_1, \ldots, r_{k+1}\}. \]
Combining this with the induction hypothesis, we conclude that
\[ \text{span}\{r_0, Ar_0, \ldots, A^k r_0, A^{k+1}r_0\} \subseteq \text{span}\{r_0, r_1, \ldots, r_{k+1}\}. \]
The above two results proves (13.2).
(13.3) can be shown for \( k + 1 \) as follows,
\[ \text{span}\{p_0, p_1, \ldots, p_k, p_{k+1}\} = \text{span}\{p_0, p_1, \ldots, p_k, r_{k+1}\} = \text{span}\{p_0, p_1, \ldots, p_k, A^k r_0, A^{k+1} r_0\} \]
Next, we show the conjugacy (13.4). From \( p_{k+1} = -r_{k+1} + \beta_k p_k \), multiplying by \( Ap_i \), \( i = 0, 1, \ldots, k \) gives
\[ p_{k+1}^T Ap_i = -r_{k+1}^T Ap_i + \beta_k p_k^T Ap_i. \] (13.5)
When \( i = k \), then \( p_{k+1}^T Ap_i = 0 \) from the definition of \( \beta_{k+1} \). For \( i < k \), we first note that from the induction hypothesis, \( p_0, p_1, \ldots, p_k \) are conjugate, and by Theorem 12.1 (expanding subspace minimization),
\[ r_{k+1}^T p_i = 0, \quad i = 0, 1, \ldots, k. \]
Also, from (13.3), for \( i = 0, 1, \ldots, k - 1 \) we have,
\[ Ap_i \in A \text{span}\{r_0, Ar_0, \ldots, A^i r_0\} = \text{span}\{Ar_0, A^2 r_0, \ldots, A^{i+1} r_0\} \subseteq \text{span}\{p_0, p_1, \ldots, p_{i+1}\}. \]
From the above two results, we deduce that
\[ r_{k+1}^T Ap_i = 0, \quad i = 0, 1, \ldots, k - 1. \]
Therefore, the first term in (13.5) becomes zero, and the second term is also zero due to the induction hypothesis. Therefore (13.4) holds for all \( k \).
Finally (without induction), for (13.1), we first note from
\[ p_i = -r_i + \beta_i p_{i-1} \]
that \( r_i \in \text{span}\{p_i, p_{i-1}\} \) for all \( i = 1, \ldots, k - 1 \), and \( r_0 \in \text{span}\{p_0\} \) by our initialization. Since \( r_{k+1}^T p_i = 0 \) for all \( i = 0, 1, \ldots, k - 1 \) for any \( k = 1, 2, \ldots, n - 1 \) from Theorem 12.1, we conclude that \( r_k^T r_i = 0 \) for \( i = 0, 1, \ldots, k - 1 \).
Note that the proof does require that \( p_0 = -r_0 \), the steepest descent direction.
3 The Conjugate Gradient Algorithm

Now we can make Algorithm 1 more economical. First, from Theorem 13.1 we know
that $p_0, p_1, \ldots, p_{n-1}$ are conjugate wrt $A$, and therefore from Theorem 12.1 we have
$$r_k^T p_i = 0, \ i = 0, 1, \ldots, k - 1.$$  
Then, from $p_k = -r_k + \beta_k p_{k-1}$, we have
$$\alpha_k = \frac{-r_k^T p_k}{p_k^T A p_k} = \frac{r_k^T r_k}{p_k^T A p_k}.$$  
Next, using Theorem 12.1 with $r_{k+1} = r_k + \alpha_k A p_k$, and (13.1) we have
$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \frac{r_{k+1}^T (r_{k+1} - r_k) / \alpha_k}{p_k^T A p_k} = \frac{r_{k+1}^T r_{k+1}}{r_k r_k}.$$  
We also use $r_{k+1} = r_k + \alpha_k A p_k$ for computing $r_{k+1}$ instead. The resulting algorithm
is shown in Algorithm 2. The total no. of FLOPs of this algorithm is now $n^2 + 5n$, removing $n^2$
operations comparing to the previous algorithm.

Algorithm 2: CG

1. **Input:** $x_0, A, b$
2. $r_0 = Ax_0 - b$, $p_0 = -r_0$, $k = 0$
3. while $r_k \neq 0$ do
   4. $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$ (n² + n FLOPs);
   5. $x_{k+1} = x_k + \alpha_k p_k$ (n FLOPs);
   6. $r_{k+1} = r_k + \alpha_k A p_k$ (n FLOPs);
   7. $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$ (n FLOPs);
   8. $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$ (n FLOPs);
   9. $k = k + 1$
10. end
11. **Output:** $x^* = x_k$.

4 Rate of Convergence

In Theorem 11.3, we showed that the CG algorithm find the solution in at most $n$ steps. In fact, when the eigenvalues of $A$ are clustered, then it converges much faster than $n$.

To see this property, we first note from Theorem 13.1 that
$$x_{k+1} = x_0 + c_0 p_0 + \cdots + c_k p_k = x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \cdots + \gamma_k A^k r_0,$$
with some proper constants $c_i$ and $\gamma_i$. We define $P_k^*(A)$ to be a polynomial of degree $k$ with coefficients $\gamma_0, \text{dots}, \gamma_k$, so that
$$P_k^*(A) = \gamma_0 I + \gamma_1 A + \cdots + \gamma_k A^k, \ x_{k+1} = x_0 + P_k^*(A) r_0.$$  
We aim to show that among all possible methods whose first $k$ steps are restricted to the Krylov space $K(r_0; k)$, Algorithm 2 performs the best for minimizing
Due to Theorem 12.1, $x^*$ is the minimizer of $\phi$, we have that 
\[
\frac{1}{2} \|x - x^*\|^2_A = \frac{1}{2} (x - x^*)^T A (x - x^*) = \frac{1}{2} x^T A x - b^T x + x^T (b - A x^*) + \frac{1}{2} (x^*)^T A x^* \\
= \phi(x) - \phi(x^*) + ((x^*)^T A - b^T) x^* = \phi(x) - \phi(x^*).
\]

Due to Theorem 12.1, $x_{k+1}$ minimizes $\phi$, and hence $\|x - x^*\|^2_A$, over the set $x_0 + \text{span}\{p_0, \ldots, p_k\} = x_0 + \text{span}\{r_0, A r_0, \ldots, A^k r_0\}$. Therefore the polynomial $P_k^*(A)$ solves the following problem over all possible polynomials of degree $k$,
\[
\min_{P_k} \|x_0 + P_k(A) r_0 - x^*\|_A.
\]

From $r_0 = A x_0 - b = A (x_0 - x^*)$, we have
\[
x_{k+1} - x^* = x_0 + P_k^*(A) r_0 - x^* = [I + P_k^*(A) A] (x_0 - x^*).
\]

On the other hand, we can write
\[
A = \sum_{i=1}^n \lambda_i v_i v_i^T
\]
for its eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_n$ and the associated orthonormal eigenvectors $v_1, v_2, \ldots, v_n$. Since the eigenvectors span $\mathbb{R}^n$,
\[
x_0 - x^* = \sum_{i=1}^n \xi_i v_i
\]
for some coefficients $\xi_i$. Any eigenvector of $A$ is also an eigenvector of $P_k(A)$, so that
\[
P_k(A) v_i = P_k(\lambda_i) v_i \quad i = 1, 2, \ldots, n.
\]

Plugging-in (13.7) into (13.6), we have
\[
x_{k+1} - x^* = \sum_{i=1}^n [1 + \lambda_i P_k^*(\lambda_i)] \xi_i v_i
\]
Using $\|z\|^2_A = z^T A z = \sum_{i=1}^n \lambda_i (v_i^T z)^2$,
\[
\|x_{k+1} - x^*\|^2_A = \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k^*(\lambda_i)]^2 \xi_i^2.
\]

The polynomial $P_k^*$ generated by the CG algorithm minimizes the LHS, and therefore,
\[
\|x_{k+1} - x^*\|^2_A = \min_{P_k} \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2
\]

\[
\leq \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \left( \sum_{j=1}^n \lambda_j^2 \right)
\]

\[
= \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x^*\|^2_A. \quad (13.8)
\]

We use the above property to show the following theorems.
Theorem 13.2. If $A$ has only $r$ distinct eigenvalues, then the CG algorithm will terminate at the solution in at most $r$ iterations.

Proof. Suppose that the eigenvalues $\lambda_i$, $i = 1, 2, \ldots, n$, of $A$ take on the $r$ distinct values $\tau_1 < \tau_2 < \cdots < \tau_r$. We define a polynomial $Q_r(\lambda)$ by

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \cdots \tau_r} (\lambda - \tau_1) \cdots (\lambda - \tau_r).$$

Then $Q_r(\lambda_i) = 0$ for $i = 1, 2, \ldots, n$, and $Q_r(0) = 1$, and therefore $Q_r(\lambda) - 1$ is a degree-$r$ polynomial with a root at $\lambda = 0$. That is, if we define the function

$$\tilde{P}_{r-1}(\lambda) = (Q_r(\lambda) - 1)\lambda,$$

then it is a polynomial of degree $r - 1$. Taking $k = r - 1$ in (13.8), we get

$$0 \leq \min_{\tilde{P}_{r-1}} \max_{1 \leq i \leq n} [1 + \lambda_i \tilde{P}_{r-1}(\lambda_i)]^2 \leq \max_{1 \leq i \leq n} [1 + \lambda_i \tilde{P}_{r-1}(\lambda_i)]^2 = \max_{1 \leq i \leq n} Q_r^2(\lambda_i) = 0.$$

That is, $\|x_r - x^*\|_A^2 = 0$ from (13.8), and therefore $x_r = x^*$.

By similar arguments, we can show the following results.

Theorem 13.3. Let $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $A$. Then,

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

Or, for $\kappa(A) = \|A\|_2 ||A^{-1}\|_2 = \lambda_n/\lambda_1$,

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

To understand this result, suppose that $A$ has $m$ large eigenvalues, where the rest $n - m$ smaller eigenvalues are clustered around 1. Defining $\epsilon = \lambda_{n-m} - \lambda_1$, this theorem tells that after $m + 1$ steps we have

$$\|x_{m+1} - x^*\|_A \approx \frac{\epsilon}{2} \|x_0 - x^*\|_A,$$

so that for small $\epsilon$, the CG iterate $x_{m+1}$ will provide a good estimate of the solution.

References