Conjugate Gradient and GMRES

Hamad El Kahza

October 30, 2020

1 Conjugate Gradient recall:

Conjugate Gradient method aims to minimize the following quadratic function:

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \tag{3}$$

1.1 Conjugate Direction method

To find the new x term in the search process, CD uses the following expression.

$$x_{[i+1]} = x_{[i]} + \alpha_{[i]} d_{[i]}. \tag{29}$$

In order to find the value of $\alpha_{[i]}$, we choose $e_{[i+1]}$ to be orthogonal to $d_{[i]}$, so that we need never step in the direction of $d_{[i]}$ again. Therefore:

$$\alpha_{[i]} = -\frac{d_{[i]}^T Ae_{[i]}}{d_{[i]}^T Ad_{[i]}} \tag{31}$$

$$= -\frac{d_{[i]}^T r_{[i]}}{d_{[i]}^T Ad_{[i]}}. \tag{32}$$

We therefore need to generate a set of A-orthogonal search directions $d_{[i]}$. A simple way to generate them is conjugate Gram-Schmidt process.

1.2 Gram-Schmidt Conjugation

Suppose we have a set of n linearly independent vectors $\mu_0, \mu_1, \ldots, \mu_{n-1}$:
set $d_{[0]} = \mu_0$ and for $i > 0$, set

$$d_{[i]} = \mu_i + \sum_{k=0}^{i-1} \beta_{ik} d_{[k]}, \tag{1}$$

where the β_{ik} are defined for $i > k$:

$$\beta_{ik} = -\frac{\mu_i^T Ad_{[k]}}{d_{[k]}^T Ad_{[k]}}$$

The limitation of using Gram-Schmidt conjugation in the method of Conjugate Directions is that:

- all the old search vectors must be kept in memory to construct each new one
- $\mathcal{O}(n^3)$ operations are required to generate the full set

1.3 Specific type of conjugate Direction

- The conjugate gradient is a specific type of the CD algorithm
- In each iteration, the new conjugate direction is a linear combination of the current gradient and the previous conjugate direction.

The CG algorithm takes the following form:

$$
\begin{aligned}
d_{[0]} &= r_{[0]} = b - Ax_{[0]} \\
\alpha_{[i]} &= \frac{r_{[i]}^T r_{[i]}}{d_{[i]}^T A d_{[i]}} \\
x_{[i+1]} &= x_{[i]} + \alpha_{[i]} d_{[i]}, \\
r_{[i+1]} &= r_{[i]} - \alpha_{[i]} A d_{[i]}, \\
\beta_{[i+1]} &= \frac{r_{[i+1]}^T r_{[i+1]}}{r_{[i]}^T r_{[i]}}, \\
d_{[i+1]} &= r_{[i+1]} + \beta_{[i+1]} d_{[i]} \quad \text{(set i = i+1, go back to compute new step } \alpha\text{)}
\end{aligned}
$$

2 Convergence of Conjugate Gradient Method

The conjugate gradient method in n iterations.

However, the calculation of the residual gradually loses acccuracy and therefore the search direction vectors lose the A orthogonality.

Krylov subspaces such as this have another pleasing property. For a fixed i, the error term has the form

$$e_{[i]} = \left(I + \sum_{j=1}^{i} \psi_j A^j \right) e_{[0]}.$$

where the coefficients ψ_i are related to the values α_i and β_i.

$$e_{[i]} = P_i(A) e_{[0]},$$

The error term can therefore be expressed as a sum of orthogonal eigenvectors.

Note: the energy norm is defined as $\|e\|_A = (e^T A e)^{1/2}$

$$
\begin{aligned}
e_{[i]} &= \sum_j \xi_j P_i(\lambda_j) v_j \\
Ae_{[i]} &= \sum_j \xi_j P_i(\lambda_j) \lambda_j v_j \\
\|e_{[i]}\|_A^2 &= \sum_j \xi_j^2 \left[P_i(\lambda_j) \right]^2 \lambda_j.
\end{aligned}
$$

Remark: CG aims to find that polynomial that minimizes this error in A norm expression. However, the convergence is as good as the worst eigenvector.

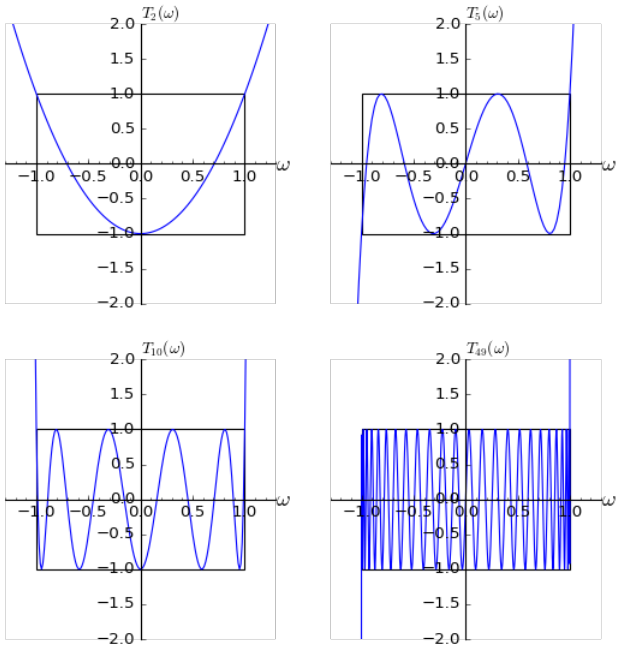
Letting $\Lambda(A)$ be the set of eigenvalues of A, we have

$$
\begin{aligned}
\|e_{[i]}\|_A^2 &\leq \min_{P_i} \max_{\lambda \in \Lambda(A)} \left[P_i(\lambda) \right]^2 \sum_j \xi_j^2 \lambda_j \\
&= \min_{P_i} \max_{\lambda \in \Lambda(A)} \left[P_i(\lambda) \right]^2 \|e_{[0]}\|_A^2. \qquad (50)
\end{aligned}
$$

2.1 Chebyshev polynomial

The Chebyshev polynomial is expressed as follows:

$$T_i(\omega) = \frac{1}{2}\left[(\omega + \sqrt{\omega^2 - 1})^i + (\omega - \sqrt{\omega^2 - 1})^i\right].$$



This method minimizes the error norm expression over the range $[\lambda_{min}, \lambda_{max}]$ by choosing:

$$P_i(\lambda) = \frac{T_i\left(\frac{\lambda_{max}+\lambda_{min}-2\lambda}{\lambda_{max}-\lambda_{min}}\right)}{T_i\left(\frac{\lambda_{max}+\lambda_{min}}{\lambda_{max}-\lambda_{min}}\right)}.$$

We enforce thew condition $P_i(0) = 1$

$$\begin{aligned}
\|e_{[i]}\|_A &\leq T_i\left(\frac{\lambda_{max}+\lambda_{min}}{\lambda_{max}-\lambda_{min}}\right)^{-1} \|e_{[0]}\|_A \\
&= T_i\left(\frac{\kappa+1}{\kappa-1}\right)^{-1} \|e_{[0]}\|_A \\
&= 2\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\right]^{-1} \|e_{[0]}\|_A. \qquad (51)
\end{aligned}$$

The second addend inside the square brackets converges to zero as i grows, so it is more common to express the convergence of CG with the weaker inequality

$$\|e_{[i]}\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i \|e_{[0]}\|_A. \qquad (52)$$

the maximum number of iterations CG requires is

$$i \leq \left\lceil \frac{1}{2}\sqrt{\kappa}\ln\left(\frac{2}{\epsilon}\right)\right\rceil.$$

CG has a time complexity of $\mathcal{O}(m\sqrt{\kappa})$. Both algorithms have a space complexity of $\mathcal{O}(m)$.

2.2 Preconditioning:

Preconditioning is usually beneficial in reducing the condition number of the Matrix we aim to solve for.

We can solve $Ax = b$ indirectly by solving

$$M^{-1}Ax = M^{-1}b. \tag{53}$$

- If $\kappa(M^{-1}A) \ll \kappa(A)$, or if the eigenvalues of $M^{-1}A$ are better clustered than those of A, we can iteratively solve the equation above more quickly than the original problem.
- However, $M^{-1}A$ is not generally symmetric nor definite, even if M and A are.
- We can find a a matrix E that satsifies the property $EE^T = M$
- The matrices $M^{-1}A$ and $E^{-1}M^{-1}E^{-T}$ have the same eigenvalues.

The system can be transformed into the problem

$$(E^{-1}AE^{-T})\widehat{x} = E^{-1}b, \qquad \widehat{x} = E^T x,$$

We obtain our transformed Preconditioned Conjugate Gradient Method:

$$\begin{aligned}
\widehat{d}_{[0]} = \widehat{r}_{[0]} &= E^{-1}b - E^{-1}AE^{-T}\widehat{x}_{[0]},\\
\alpha_{[i]} &= \frac{\widehat{r}_{[i]}^T\widehat{r}_{[i]}}{\widehat{d}_{[i]}^T E^{-1}AE^{-T}\widehat{d}_{[i]}},\\
\widehat{x}_{[i+1]} &= \widehat{x}_{[i]} + \alpha_{[i]}\widehat{d}_{[i]},\\
\widehat{r}_{[i+1]} &= \widehat{r}_{[i]} - \alpha_{[i]}E^{-1}AE^{-T}\widehat{d}_{[i]},\\
\beta_{[i+1]} &= \frac{\widehat{r}_{[i+1]}^T\widehat{r}_{[i+1]}}{\widehat{r}_{[i]}^T\widehat{r}_{[i]}},\\
\widehat{d}_{[i+1]} &= \widehat{r}_{[i+1]} + \beta_{[i+1]}\widehat{d}_{[i]}.
\end{aligned}$$

Constraints:

- E must be computed $\rightarrow$ not desirable
- Setting $\widehat{r}_{[i]} = E^{-1}r_{[i]}$ and $\widehat{d}_{[i]} = E^T d_{[i]}$, and using the identities $\widehat{x}_{[i]} = E^T x_{[i]}$ and $E^{-T}E^{-1} = M^{-1}$ we obtaine the untransformed preconditioned CG

Untransformed preconditioned CG

$$
\begin{aligned}
r_{[0]} &= b - Ax_{[0]}, \\
d_{[0]} &= M^{-1} r_{[0]}, \\
\alpha_{[i]} &= \frac{r_{[i]}^T M^{-1} r_{[i]}}{d_{[i]}^T A d_{[i]}}, \\
x_{[i+1]} &= x_{[i]} + \alpha_{[i]} d_{[i]}, \\
r_{[i+1]} &= r_{[i]} - \alpha_{[i]} A d_{[i]}, \\
\beta_{[i+1]} &= \frac{r_{[i+1]}^T M^{-1} r_{[i+1]}}{r_{[i]}^T M^{-1} r_{[i]}}, \\
d_{[i+1]} &= M^{-1} r_{[i+1]} + \beta_{[i+1]} d_{[i]}.
\end{aligned}
$$

3 GMRES

The Arnoldi iteration is used to solve systems. The algorithm for this matter is called GMRES.

3.1 Minimization of the residual in $\mathcal{K}_n$

We try to solve for $x_* = A^{-1}b$.

The idea of the algortithm is at the iteration n, We approximate x_* by the vector $x_n \in \mathcal{K}_n$ which minimizes the norm of the residual $r_n = b - Ax_n$.
The vector $x \in \mathcal{K}_n$ is written as

$$x = c_0 b + c_1 Ab + \cdots + c_{n-1} A^{n-1} b = q(A)b\,,$$

where $q(z) = \sum_{i=0}^{n-1} c_i z^i$. The residual is therefore written as $r_n = b - Ax_n = p_n(A)b$, where $p_n \in P_n$, where P_n is the set of polynomials p of degree larger or equal to n as for $p(0) = 1$. We choose $p_n \in P_n$ as per $p_n(z) = 1 - zq(z)$.

- x belongs to K_m is equivalent to x= K_mc

Other methods to write this expression

$$
\begin{aligned}
&\min ||r_n|| \\
&||b - Ax_n|| \\
&||AK_n c - b|| \\
&||p_n(A)b|| \\
&||AQ_n y - b||
\end{aligned}
$$

The last line is immediate recalling the subspace underlying the columns K_n is similar to the space underlying the columns of Q.

We therefore solve a linear system $m \times n$. We use the Arnoldi iteration $AQ_n = Q_{n+1}\tilde{H}_n$ to reduce the size.

Algorithm 1 Algorithm ARNOLDI

Data: $\|q_1\|$=1
  Result: Krylov subspace
  begin
    $q_k \leftarrow Aq_{k-1}$
  for $j = 1, 2, 3, \ldots, k-1$ **do**
    $h_{j,k-1} \leftarrow q_j^T q_k$
    $q_k = q_k - q_j^T h_{j,k-1} q_j$
    $h_{k,k-1} \leftarrow ||q_k||$
    $q_k = q_k / h_{k,k-1}$
    end
  return q_k
  end

1. We start with an arbitrary vector q_1 with a norm equal to 1, so we may obtain an orthonormal basis
2. We apply matrix A to q_{k-1} to obtain the next iterate q_k
3. We iterate over j $\rightarrow$ k-1 to find the h vector which defines the length of the projection onto q
4. q_k is assigned the value q_k minus the legth of the projection
5. set $h_{k,k-1}$ to the norm of q_k
6. normalize q_k to $\|q_k\|$

The problem to resolve is therefore

$$\|Q_{n+1}\tilde{H}_n y - b\|_{=}\text{minimum.}$$

As the two vectors in the norm are in the space of the columns of Q_{n+1}, multiplying the left side by Q_{n+1}^* does not change this norm. We can also minimize this norm $\|\tilde{H}_n y - Q_{n+1}^* b\|$. We therefore write the final form to solve the least square problem GMRES GMRES:

$$\|\tilde{H}_n y - \|b\|_{e1}\|_{=}\text{minimum.}$$

At n^{th} iteration, we resolve the problem to find y, and subsequently $x_n = Q_n y$. This new problem has a dimension of $(n+1) \times n$.

3.2 Algorithm gmres

Algorithm 2 Algorithm GMRES

Data: The problem $Ax = b$ is Symmetric definitive positif.
Result: A solution approaching, x_n.
begin
  $q_1 \leftarrow b/\|b\|$ **end**
  for $n = 1, 2, 3, \ldots$ **do**
    Find $\tilde{H}_n$ using Arnoldi Find y that minimizes $\|\tilde{H}_n y - \|b\|_{e1}\|$
    $x_n \leftarrow Q_n y$
  return x_n
end

3.3 gmres and the polynomial approximation

The iterate is written as

$$x_n = q_n(A)b\,,$$

The residual is $r_n = b - Ax_n$ est $r_n = (I - Aq_n(A))b$. Therefore, we can say

$$r_n = b - Aq_n(A)b = p_n(A)b$$

For a certain polynomial $p_n \in P_n$ as for $p_n(z) = 1 - zq(z)$. GMRES chooses the coefficients of the polynomial p_n to minimize the norm residual and resolve the next approximation problem.

[Approximation of GMRES*] Find* $p_n \in P_n$ *as for* $\|p_n(A)b\|$ *is minimum.*

3.4 Convergence of gmres

The algorithm converges monotonically:

$$\|r_{n+1}\| \leq \|r_n\|\,, \quad \|p_{n+1}(A)b\| \leq \|p_n(A)b\|\,.$$

- It can be seen for $\|r_n\|$ is the smallest possible for $\mathcal{K}_n$,
- For enlarging the subspace to $\mathcal{K}_{n+1}$, The optimal solution at the new subspace can only minimize the residual
- In exact arithmetic, the algorithm must converge at most at m'th iterate $\|r_n\| = 0$

The convergence criteria deduces to:

$$\|r_n\| = \|p_n(A)b\| \leq \|p_n(A)\|_{|}|b||,$$

Therefore *the rate of convergence* can be written as:

$$\frac{||r_n||}{||b||} \leq \inf_{p_n \in P_n} \|p_n(A)\|$$