Kafa Alameh

July 31, 2018

1 Krylov subspace methods

Krylov subspace methods are polynomial iterative methods that aim to solve linear equations of the form $Ax = b$ where A is a known $n \times n$ matrix, b is a known $n \times 1$ vector and x is an $n \times 1$ vector of unknowns. The following are Krylov projection methods. These methods impose a constraint of the residuals by requiring them to form an orthogonal set and the A-orthogonality of the search directions, except for MSD-CG.

1.1 Conjugate Gradient (CG)

An iterative Krylov projection method restricted for symmetric positive definite (SPD) (Hermitian) matrices:

- $Ax = b$
- $A = A^T$
- $x^T Ax > 0 \; \forall x \neq 0$

Given an initial guess or iterate x_0, at the k^{th} iteration CG generates $\{x_k\}$ by minimizing a function $\phi(x)$. CG finds an approximate solution $x_k = x_{k-1} + \alpha_k p_k$, where p_k is a vector, the search direction, α_k is a scalar determining the step length.
Minimizing $\phi(x) = \frac{1}{2}x^T Ax - b^T x$ ($\nabla\phi(x) = 0$) is equivalent to solving $Ax = b$.
The convergence criterion is set as $||r_k||_2 \leq \epsilon||b||_2$, for some $\epsilon \in \Re$.
Input:

- SPD $n \times n$ matrix A, $n \times 1$ vector b
- initial guess or iterate x_0
- stopping tolerance ϵ, the maximum allowed iterations k_{max}

Output:

- the approximate solution x_k

The Algorithm:

1. Start with some x_0. Set $p_0 = r_0 = b - Ax_0$. Set $\rho_0 = ||r_0||_2^2$. Set $k = 0$
2. while ($\sqrt{\rho_{k-1}} > \epsilon||b||_2$ and $k < k_{max}$) do
3. $\quad x_{k+1} = x_k + \alpha_k p_k$, $\alpha_k = \frac{r_k^T r_k}{p_k^T}$
4. $\quad r_{k+1} = b - Ax_{k+1} = r_k - \alpha_k A p_k$
5. $\quad p_{k+1} = r_{k+1} + \beta_k p_k$, $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
6. $\quad \rho_k = ||r_{k+1}||_2^2$
7. end while

1.2 Block conjugate gradient (B-CG)

Similar to CG, B-CG solves a SPD system with multiple right-hand sides:

- $AX = B$
- $A = A^T$
- $x^T Ax > 0 \ \forall x \neq 0$

where A is an $n \times n$ matrix, $X \in \Re^{n \times t}$ is a block vector and B is a block vector of size $n \times t$ containing multiple right hand sides.
Input:

- SPD $n \times n$ matrix A, $n \times t$ block B of t right hand sides
- block of t initial guesses or iterates X_0
- stopping tolerance ϵ, the maximum allowed iterations k_{max}

Output:

- the block of t approximate solutions X_k

The Algorithm:

1. Start with some X_0. Set $= R_0 = B - AX_0$. Set $\rho_0 = ||R_0||_2^2$. Set $k = 0$
2. while ($\sqrt{\rho_{k-1}} > \epsilon||B||_2$ and $k < k_{max}$) do
3. $\quad$ if($k = 1$) then let $P = R_0$
4. $\quad$ else let $P = R + P\beta$

5. orthogonalize the vectors of P against each other and define γ as a $t \times t$ full rank freely chosen matrix

6. end if

7. $X_{k+1} = X_k + \alpha_{k+1} P_{k+1}$, $\alpha_{k+1} = (P_{k+1}^T A P_{k+1})^{-1} \gamma_k^T R_k^T R_k$

8. $R_{k+1} = R_k - A P_{k+1} \alpha_{k+1}$

9. $P_{k+2} = (R_{k+1} + P_{k+1}\beta_{k+2})\gamma_{k+2}$, $\beta_{k+2} = \gamma_{k+1}^{-1}(R_k^T R_k)^{-1} R_{k+1}^T R_{k+1}$

10. $\rho_k = ||R_{k+1}||_2^2$

11. end while

1.3 S-step conjugate gradient

A parallelizable version of Krylov methods where s iterations of classical Krylov methods are merged and computed simultaneously. Instead of one iteration at a time, the iterations are performed in blocks of s. We have s directions p_k so that we can write:
$x_{k+s} = x_k + \alpha_k p_k + ... + \alpha_{k+s-1} p_{k+s-1} = x_k + P_k a_k$
where $P_k = [p_k, ..., p_{k+s-1}]$ and $a_k = [\alpha_k, ..., \alpha_{k+s-1}]$
The residual can be expressed as $r_{k+s} = r_k - A P_k a_k$
The next direction is expressed as a combination of previous directions and the basis vectors since we have the basis for Krylov subspace for s iterations: $B_k = [r_k, Ar_k, ..., A^{s-1} r_k]$
$P_{k+1} = B_k + P_k \beta_k$, where β_k is a $s \times s$ matrix. As before, we get expressions for α_k and β_k by enforcing orthogonality of residual and search directions.
$\alpha_k = (P_k^T A P_k)^{-1} P_k^T r_{k-1}$ and $\beta_k = (P_k^T A P_k)^{-1} P_k^T A B_k$.
Input:

- SPD $n \times n$ matrix A, $n \times 1$ vector b
- initial guess or iterate x_0, the number of steps per iteration s
- stopping tolerance ϵ, the maximum allowed iterations k_{max}

Output:

- the approximate solution x_k

The algorithm is the same as before but with an additional definition of $P = [r, Ar, ..., A^{s-1} r]$ at the beginning and the definition of $B = [r, Ar, ..., A^{s-1} r]$ in the while loop.

1.4 Cooperative-CG (coop-CG)

Similar to block conjugate gradient, coop-CG make all search directions conjugate to each other. Coop-CG solves the system $Ax = b$ by starting with t distinct initial guesses. This is equivalent to solving the system $AX = b * ones(1, t)$. Given X_0 as a the vector containing t initial guesses, the block residual is given by $R_0 = AX_0 - b * ones(1, t)$.

The algorithm is exactly the same as B-CG with $\gamma_k = I$. The convergence criterion is also defined as before. However, since we have a block of residuals, ρ is defined as $\rho = min(||R_0(:,1)||_2^2, ||R_0(:,2)||_2^2, ..., ||R_0(:,t-1)||_2^2, ||R_0(:,t)||_2^2)$.

1.5 Multiple search directions CG (MSD-CG)

MSD-CG solves the system $Ax = b$ by partitioning A's domain into t subdomains and defining a search direction on each of the t subdomains. MSD-CG does not have the A-orthogonality of the search domains. However, β_k is chosen such that the global search direction p^k is A-orthogonal to the previous domain search direction p^{k-1}, i.e. $(p^k)^T AP_{k-1} = 0$, for $i = 1, 2, ..., t$. At each iteration k, a search direction p_i^k is efined on each of the t subdomains $(\delta_i, i = 1, 2, ..., t)$ such that $p_i^k(\delta_j) = 0$ for all $j \neq i$. Given a matrix containing all the search directions $P_k = [p_1^k, p_2^k, ..., p_t^k]$ and the vector α_k of size t, at the k^{th} iteration, the approximate solution is defined as $x_k = x_{k-1} + P_k\alpha_k$. α_k and β_k are defined as $\alpha_k = (P_k^T AP_k)^{-1}P_k^T r_{k-1}$ and $\beta_k = (P_{k-1}^T AP_{k-1})^{-1}P_{k-1}^T Ar_{k-1}$.