Data Sparse Matrix Computation
Lecture 8: Krylov Subspace Methods

Yao Cheng

November 7, 2017

Contents

1 Introduction

In last lecture, we gave a brief introduction to krylov subspace methods. The basic setting of this kind of methods is as follows:

1. We have a black box which computes Ax and returns it.
2. We build an iterative method which generates a sequence $x^{(k)} \to x$ with $Ax = b$ as $k \to \infty$.
3. We decide to consider

$$x^{(k)} \in \mathcal{K}_k(A, b) = span\{b, Ab, ..., A^{(k-1)}b\}$$

The subspace mentioned above is krylov subspace, which has the following properties:

1. It can be constructed just with the black box
2. $\mathcal{K}_k(A, b) \subseteq \mathcal{K}_{k+1}(A, b)$
3. It motivates considering $x^{(k)} = P_k(A)b$

Krylov sequences $[b, Ab, ..., A^{k-1}b]$ forms a basis for Krylov subspace but it is ill-conditioned. It is better to work with an orthonormal basis.

Next we will introduce two algorithms to build orthonormal basis.

2 Arnoldi algorithm

2.1 Hessenberg Reduction

Given a $n \times n$ matrix A, we can compute an orthogonal matrix Q and an upper Hessenberg matrix H(upper triangular and one sub-diagonal) s.t. $A = QHQ^*$.

For iterative methods, we take the view that n is huger of infinite. Thus instead of considering the full Q, we consider the first k column of $AQ = QH$.

Let Q_k be the first k columns of matrix Q and $\hat{H}_k$ the upper left $(k+1) \times k$ block of H. That is,

$$Q_k = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ q_1 & q_2 & \dots & q_k \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \hat{H}_k = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,k} \\ h_{2,1} & h_{2,2} & \dots & \vdots \\ 0 & h_{3,2} & \dots & \vdots \\ 0 & 0 & \vdots & h_{k,k} \\ 0 & 0 & 0 & h_{k+1,k} \end{bmatrix}$$

Thus $AQ_k = Q_{k+1}\hat{H}_k$, which means that

$$Aq_k = h_{1,k}q_1 + h_{2,k}q_2 + ... + h_{k,k}q_k + h_{k+1,k}q_{k+1}$$

That is, q_{k+1} satisfies an $(k+1)$-term recurrence relation involving itself and previous Krylov vectors.

Therefore, "h"s just correspond to modified Gram-Schmidt orthogonalization. And Arnoldi algorithm is simply the modified Gram-Schmidt iteration that implements the above equation.

2.2 Arnoldi Algorithm

The following is Arnoldi algorithm:

Arnoldi Algorithm:
Initialize $\mathbf{b}$ as a random vector, $\mathbf{q_1} = \frac{\mathbf{b}}{||\mathbf{b}||_2}$
for $k = 1, 2, ...$ **do**
  $\mathbf{v} = A\mathbf{q_k}$
  for $j = 1, 2, ...k$ **do**
    $h_{jk} = \mathbf{q_j^*v}$
    $\mathbf{v} = \mathbf{v} - h_{jk}\mathbf{q_j}$
  end for
  $h_{k+1,k} = ||\mathbf{v}||_2$
  $\mathbf{q_{k+1}} = \mathbf{v}/h_{k+1,k}$
end for

Given this algorithm, $Q_k = [\mathbf{q_1}, \mathbf{q_2}, ..., \mathbf{q_k}]$ is an orthonormal basis for $\mathcal{K}_n(A, b)$. Note that $AQ_k = Q_{k+1}\hat{H}_k$ where $\hat{H}_k$ is $\begin{bmatrix} H_k \\ h_{k+1,k}e_k^T \end{bmatrix}$.

Thus we have $Q_k^* A Q_k = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} H_k \\ h_{k+1,k}e_k^T \end{bmatrix}$ =H$_k$, where H_k is tridiagonal and can be interpreted as the representation in the basis $\{\mathbf{q_1}, \mathbf{q_2}, ..., \mathbf{q_k}\}$ of the orthogonal projection of A onto $\mathcal{K}$.

3 Lanczos Algorithm

Lanczos algorithm builds an orthonormal basis for Krylov subspace for hermitian matrix(that is, a complex square matrix that is equal to its own conjugate transpose). The following is Lanczos algorithm:

Lanczos Algorithm:
Given $A = A^*$,initialize $\mathbf{b}$ as a random vector, $\beta_0 = 0, \mathbf{q_0} = 0$,$\mathbf{q_1} = \frac{\mathbf{b}}{||\mathbf{b}||_2}$
for $k = 1, 2, ...$ **do**
  $\mathbf{v} = A\mathbf{q_k}$
  $\alpha_k = \mathbf{q_k}^T\mathbf{v}$
  $\mathbf{v} = \mathbf{v} - \beta_{k+1}\mathbf{q_{k+1}} - \alpha_k\mathbf{q_k}$
  $\beta_k = ||\mathbf{v}||_2$
  $\mathbf{q_{k+1}} = \mathbf{v}/\beta_k$
end for

If we define T_k= $\begin{bmatrix} \alpha_1 & \beta_1 & 0 & \dots & \dots & 0 \\ \beta_1 & \alpha_2 & \beta 2 & \dots & \dots & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \beta_{k-1} & \alpha_k \end{bmatrix}$ Then we have $AQ_k = Q_k\hat{T_k}$ where $\hat{T_k} = \begin{bmatrix} T_k \\ \beta_k e_k^T \end{bmatrix}$

$\forall k$, q_k is a three-term recurrence relation involving itself and previous Krylov vectors, which is computational efficient.

4 Solving the system in coordinate space

Having defined the basis for Krylov subspace, we want to solve $Ax = b$ with $x^{(k)} \in \mathcal{K}_k(A, b)$ and 0 as our initial guess.

Thus $x^{(k)}$ should be the "best" vector in $\mathcal{K}_k(A, b)$ where "best" means $x^{(k)} = \text{argmin}_{x \in \mathcal{K}_k(A,b)} ||Ax - b||_2^2$. This can be done via MINRES[1] if $A = A^*$ or via GMRES[2] in more general cases.

What if we want to solve $x^{(k)} = \text{argmin}_{x \in \mathcal{K}_k(A,b)} ||x - A^{-1}b||_2^2 = x^T x + b^T A^{-T} A^{-1} b - x^T A^{-1} b - b^T A^{-T} x$? The problem is that A^{-1} cannot be eliminated.

5 Conjugate Gradient

Conjugate gradient is a kind of Krylov space solver that only applies to systems that are symmetric positive definite(spd).

The goal of conjugate gradient is to solve the problem $x^{(k)} = \text{argmin}_{x \in \mathcal{K}_k(A,b)} ||x - A^{-1}b||_A^2$ where A is spd.

This can be written as $\text{argmin}_{y \in \mathbb{R}^k} ||Q_k y - A^{-1}b||_A^2 = \text{argmin}_{y \in \mathbb{R}^k} y^T Q_k^T A Q_k y + b^T A^{-1} b - 2y^T Q_k^T b = \text{argmin}_{y \in \mathbb{R}^k} y^T T_k y - 2y^T e_1 ||b||_2$

Take derivative and set it equal to zero, we will have $T_k y - ||b||_2 e_1 = 0$, that is, $T_k y = ||b|| e_1$.

In conclusion, when solving $Ax = b$ using Krylov subspace method, we run Lanczos with A,b at each step to get Q_k and T_k. Then we solve $Q_k y^{(k)} = ||b||_2 e_1$ and set $x^{(k)} = Q_k y^{(k)}$.

References

[1] Christopher C Paige and Michael A Saunders. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis*, 12(4):617–629, 1975.

[2] Youcef Saad and Martin H Schultz. Conjugate gradient-like algorithms for solving nonsymmetric linear systems. *Mathematics of Computation*, 44(170):417–424, 1985.