Iterative methdos

Zecheng Zhang

April 24, 2023

1 Eigenvalue problem

We will consider symmetric matrix $A \in \mathbb{R}^{m\times m}$. We define the Rayleigh Quotient,

$$r(x) = \frac{x^t Ax}{x^t x}. \tag{1}$$

Note that if x is an eigenvector of A, $r(x) = \lambda$ is its eigenvalue.

One way to understand this formula is: given x, what is the scale α which acts almost like an eigenvalue of x in the sense that $Ax - \alpha x$ is minimized? This is a least square problem, but x is the matrix α is the unknown vector, and Ax is the right-hand side b vector. We can see that $\alpha = r(x)$ if we consider the normal equation.

Take the derivative of $r(x)$ with respect to all component x_j of x, we can easily derive that,

$$\nabla r(x) = \frac{2}{x^t x}(Ax - r(x)x). \tag{2}$$

We can see that when x is the eigenvector, the gradient vanishes. Conversely, if the gradient is trivial with $x \neq 0$, x is an eigenvector with eigenvalue $r(x)$.

Theorem 1.1. Let q_j be an eigenvector of A, we have

$$r(x) - q_j = \mathcal{O}(\|x - q_j\|^2), \tag{3}$$

as $x \to q_j$.

The Power iteration is expected to return an eigenvector corresponding to the largest eigenvalues.

Algorithm 1: Power Iteration

```
Set v_0 with ||v_0|| = 1.
for k = 1 to ... do
    w = A v^k
    v^k = w / ||w||
    λ^k = (v^k)^T A v^k
```

Theorem 1.2. Suppose $|\lambda_1| > |\lambda_2| \geq ... \geq |\lambda_m| \geq 0$ and $q_1^T v^0 \neq 0$. Then the algorithm satisfies,

$$\|v^k - q_1\| = \mathcal{O}(\left|\frac{\lambda_2}{\lambda_1}\right|^k), \tag{4}$$

$$|\lambda^k - \lambda_1| = \mathcal{O}(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}), \tag{5}$$

as $k \to \infty$

Remark 1. Power iteration has some limitations.

1. It can only find the largest eigenvectors corresponding to the largest eigenvalues.
2. The convergence is linear, i.e., the algorithm reduces the error by a factor $|\frac{\lambda_2}{\lambda_1})$ in every iteration.
3. The quality of the convergence depends on the quotient. If there is no huge eigen-gap, the convergence is slow.

1.1 Inverse Iteration

Let μ be a number which is not an eigenvalue of A, the eigenvectors of $(A-\mu I)^{-1}$ are the same as the eigenvectors of A, and the coresponding eigenvalues are $(\lambda_j - \mu)^{-1}$, where $\{\lambda_j\}$ are the eigenvalues of A.

This motivates us to design an algorithm to identify λ_j and the corresponding eigenvectors of A. Suppose we know any estimate of λ_j and denote it as μ. $(\mu-\lambda_j)^{-1}$ will be very large. According to the Remark, the power iteration can identify q_j, which are the eigenvectors of $(A-\mu I)^{-1}$ (also the eigenvectors of A). This idea is called the inverse iteration.

Algorithm 2: Inverse iteration

$v^0 =$ some vectors with norm 1
for $k = 1$ *to* ... **do**
  Solve $(A-\mu I)w = v^{k-1}$ for w
  $v^k = w/\|w\|$
  $\lambda^k = (v^k)^T A v^k$.

Rayleigh quotient is one method to estimate eigenvalues from an eigenvector estimation. Inverse iteration is an estimate of the eigenvector from the eigenvalues.

Algorithm 3: RQ iteration

$v^0 =$ some vectors with norm 1
$\lambda^0 = v^0 A v^0 =$ coresponding Rayleigh quotient.
for $k = 1$ *to* ... **do**
  Solve $(A-\lambda^{k-1} I)w = v^{k-1}$ for w
  $v^k = w/\|w\|$
  $\lambda^k = (v^k)^T A v^k$.

Without proof, the Rayleigh Quotient iteration has cubic convergence.

2 Reduction to Hessenberg form

Schur factorization returns $A = QTQ^*$, where T is a triangular matrix, i.e., we would like to apply unitary similarity transformation to introduce zeros below the diagonal. The natural first idea is to use the Householder.

The first Householder reflector Q_1^* multiplied on the left of A would introduce zeros below the diagonal in the first column, and the Householder reflector will change all rows of A. This is good up to now; however, if we complete the process of multiplying Q_1 on the right, all zeros previously introduced are destroyed. We will verify this in class.

The good idea in step 1 is to choose a unitary matrix Q_1^* that will leave the first row unchanged. It will change the second row to the last row and introduce zeros below the second entry in the first column. It can be verified that the right multiplication by Q_1 will not change the zeros introduced by Q_1^*. After repeating this process for $m-2$ times, the resulting matrix is in the Hessenberg form, denoted as H.

Algorithm 4: Reduction to Hessenberg

for $k = 1$ *to* $m-2$ **do**
- $x = A_{k+1:m,k}$
- $v_k = (sign(x_1))\|x\|_2 e_1 + x$
- $v_k = v_k/\|v_k\|$
- $A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k v_k^* A_{k+1:m,k:m}$
- $A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2A_{1:m,k+1:m} v_k v_k^*$

When A is Hermitian, H is symmetric, then H is a tridiagonal matrix.

3 QR Algorithm

Algorithm 5: QR Algorithm

$A_1 = A$

for $k = 1$ *to* ... **do**
- $Q_k R_k = A_k$
- $A_{k+1} = R_k Q_k$.

The algorithm converges to the Schur form of the matrix A. Specifically, suppose A admits the Schur decomposition $A = UTU^T$, then A_k converges to T.

Remark 2. Some properties regarding the algorithm.

1. $A_{k+1} = R_k Q_k$, since $A_k = Q_k R_k$, $R_k = Q_k^t A_k$, this implies that $A_{k+1} = Q_k^t A_k Q_k$. That is, all A_k are unitarily similar to each other, i.e., eigenvalues of all A_k and A are the same. Since A^k converges to T, we have the eigenvalues of A.

2. Let us define $Q^{(k)} = Q_1 Q_2 ... Q_k$ and $R^{(k)} = R_k R_{k-1} ... R_1$, we have the following theorem.

   Property 3.0.1. (a) $A_{k+1} = (Q^{(k)})^t A Q^{(k)}$.
   (b) $A^k = Q^{(k)} R^{(k)}$.

   Proof. The property (a) is trivial to prove and let us the property (b). Let us prove by induction. $k = 1$ case is trivial. Suppose $A^{k-1} = Q^{(k-1)} R^{(k-1)}$ is true. By the property (a) and the algorithm definition, we have,

$$A_k = (Q^{(k-1)})^t A Q^{(k-1)} = Q_k R_k. \tag{6}$$

Multiplying both sides by $Q^{(k-1)}$, it follows that $AQ^{(k-1)} = Q^{(k-1)}Q_kR_k$. Substitute into the assumption,

$$A^k = AA^{k-1} = AQ^{(k-1)}R^{(k-1)} = Q^{(k-1)}Q_kR_kR^{(k-1)} = Q^{(k)}R^{(k)}. \tag{7}$$

□

The property provides us with one way to compute the QR factorization of matrix power. It can be shown that, this algorithm is stable.

We now intuitively explain the connection between QR and the Power iteration. It can be shown that columns of A^k are dominated by the "leading" eigenvector x_1 of A, i.e., $Ax_1 = \lambda_1 x_1$. Let us consider $A^k e_1 = Q^{(k)}R^{(k)}e_1 = cq_1$, where q_1 is the first column of $Q^{(k)}$ scaled by constant c. This implies that the leading eigenvector of A is related to q_1. Property (a) shows that $A_{k+1} = (Q^{(k)})^t AQ^{(k)}$ and A_{k+1} is the Schur form of A, this indicates that q_1 is the eigenvector of A and $A_{k+1}[1,1]$ is the corresponding eigenvalue.

3.1 Shifted QR

Algorithm 6: Shifted QR Algorithm

```
A_1 = A
for k = 1 to ... do
    Q_k R_k = A_k − s_k I
    A_{k+1} = R_k Q_k + s_k I.
```

If $s_k \sim \lambda_n$, then $A_{k+1}[m,m] \sim \lambda_m$. It can be shown that $(A - s_kI)(A - s_kI)...(A - s_kI) = Q^{(k)}R^{(k)}$.

3.2 Preprocessing

For QR and shifted QR, we need to run Householder to QR the matrix A_k in each iteration. The cost is m^3 for one QR, this is very costly. It is important to find a good initial condition to reduce the number of iterations.

As we have discussed before, $A_k[m,m]$ converges to λ_m. Motivated by the inverse iteration, the iterative algorithm will find it very fast if we choose s_k closed to λ_m. We can choose $s_k = A_k[m,m]$ or some number that is close to $A_k[m,m]$.

One method that works well is to reduce the matrix A to the Hessenberg form. Hessenberg form is different from the Schur form, but it is very close to the upper triangular form.

4 Iterative methods

In this section, let us consider matrix $A \in \mathbb{R}^{m\times m}$. The iterative methods has a structure $x_{n+1} = \phi(x_n)$, where x_n is the output of $n-$ step and ϕ is the algorithm. Broadly speaking, the idea of iterative methods is to:

1. Gradually refine the solution iteratively.

2. Each iteration should be (a lot) cheaper than direct methods.

3. Iterative methods can be (but not always) much faster than direct methods.

4. Tends to be (slightly) less robust, nontrivial/problem-dependent analysis. After (n^3) steps, it often gets the exact solution (ignoring roundoff errors). But one would hope to get an acceptably good solution long before that

The big idea behind Krylov subspace methods is to approximate the solution in terms of a polynomial of the matrix times a vector. Namely, in Krylov subspace methods, we look for an (approximate) solution of the form

$$p_{k-1}(A)v, \tag{8}$$

where p_{k-1} is a polynomial of degree at most $k-1$, v is the initial vector. Here $p_{k-1}(A) = \sum_{i=0}^{k-1} c_i A^i$ for some coefficients $c_i \in \mathbb{R}$.

One example is the Power method. We represent the eigenvector of A as $A^{k-1}v$, which is a special case of $p_{k-1}(A)$.

Now the goal is to find an approximation solution $\hat{x} = p_{k-1}(A)b$ in Krylov subspace

$$K_n(A, b) = span\{b, Ab, A^2b, ..., A^{n-1}b\}. \tag{9}$$

You would want to convince yourself that any vector in the Krylov subspace can be written as a polynomial of A times the vector b. The claim can be verified very easily. let $v \in K_n$, i.e., $v = \sum_{i=0}^{n-1} c_i A^i b = b \sum_{i=0}^{n-1} c_i A^i$. Let $p(z) = \sum_{i=0}^{n-1} c_i z^i$, we are done.

An important and non-trivial step towards finding a good solution is to form an orthonormal basis for the Krylov subspace, or we want to find $\{q_1, ..., q_n\}$ which is a set of orthonormal vectors which span the same space as K_n.

Algorithm 7: Arnoldi Iteration

Set up b and $q_1 = b/\|b\|$.
for $n = 1$ *to* ... **do**
    $v = Aq_n$.
    for $j = 1$ *to* n **do**
        $h_{jn} = q_j^T v$,
        $v = v - h_{jn} q_j$.
    $h_{n+1,n} = \|v\|$,
    $q_{n+1} = v/h_{n+1,n}$.

We have remarks regarding the algorithm. Firstly, we can see that $span\{b, Ab, A^2b, ..., A^{k-1}b\} = span\{q_1, ..., q_{k-1}\}$. Secondly, at $k-$th step, we have

$$Aq_n - h_{1n}q_1 - h_{2n}q_2 - ... - h_{nn}q_n = h_{n+1,n}q_{n+1}, \tag{10}$$

or we have,

$$Aq_n = h_{1n}q_1 + h_{2n}q_2 + ... + h_{nn}q_n + h_{n+1,n}q_{n+1}. \tag{11}$$

We can write it in the matrix form. Specifically, we have,

$$A[q_1, ..., q_n] = [q_1, ..., q_{n+1}] \underbrace{\begin{pmatrix} h_{11} & ... & h_{1n} \\ h_{21} & ... & h_{2n} \\ ... & ... & ... \\ 0 & h_{n,n-1} & h_{nn} \\ 0 & ... & h_{n+1,n} \end{pmatrix}}_{\tilde{H}_n}. \tag{12}$$

Here $\tilde{H}_n \in \mathbb{R}^{n+1,n} \in \mathbb{R}^{(n+1)\times n}$, note that the upper section of this matrix is a Hessenberg matrix. Let use further denote $Q_n = [q_1, ..., q_n] \in \mathbb{R}^{m\times n}$ and $Q_{n+1} = [q_1, ..., q_{n+1}] \in \mathbb{R}^{m\times(n+1)}$.

$h_{n+1,n}$ may be equal to 0, this is called a breakdown of the Arnoldi iteration, but it is a breakdown of a benigh sort. For the computaion of the eigenvalue and solving system of equations, the breakdown means that convergence has happened, and iteration terminates. Alternatively, a new orthonormal vector q_{n+1} could be selected at random.

We can write equation 11 in another form. Specifically,

$$AQ_n = Q_n \underbrace{\begin{pmatrix} h_{11} & ... & h_{1n} \\ h_{21} & ... & h_{2n} \\ ... & 0 & h_{n-1,n} \\ 0 & h_{n,n-1} & h_{nn} \end{pmatrix}}_{H_n} + [0, 0, ..., q_{n+1}][0, ..., h_{n+1,n}e_{n+1}] \tag{13}$$

Here H_n is a square matrix and is called the Hessenberg matrix. It is not hard to see $Q_n^t[0, ..., q_{n+1}] = 0$, this implies that $Q_n^t A Q_n = H_n$. Consequently, if A is symmetric, H_n is a tri-diagonal matrix.

We have discussed that any vector in the Krylov subspace can be written as $p(A)b$ for some polynomial p, the ierative method can be analyzed as finding the polynomial of A. We here define the Arnoldi approximation problem.

Definition 4.1. Find $p^n \in P^n$ such that,

$$\|p^n(A)b\| \tag{14}$$

is minimized. Here $P^n(\cdot)$ is the monic polynomials of degree n.

Theorem 4.2. As long as Arnoldi iteration does not break down (K_n is of full rank), the Arnoldi approximation problem has a unique solution p^n, this polynomial is the characteristic polynomial of H_n.

Proof. For $p \in P^n$, $p(A)b \in K_{n+1}$, consequently, $p(A)b = A^n b - Q_n y$ for some y. This turns to be a least square problem: find y such that $\|A^n - Q_n y\|$ is minimized, or, find points in K_n closest to $A^n b$. The solution satisfies that $p(A)b$ be orthogonal with Q_n, or, $0 = Q_n^* p(A)b$.

Let us now factor $A = QHQ^*$. At $n-$th step, we have computed the first n columns of Q and H. There exist $U \in\in \mathbb{R}^{m\times(m-n)}$ with orthonormal columns and satisfies $Q_n^* U = 0$ and some other matrices X_1, X_2 with upper right entry equal to 0, and X_3 such that,

$$Q = [Q_n, U], \tag{15}$$

and

$$H = \begin{bmatrix} H_1 & X_1 \\ X_2 & X_3 \end{bmatrix}. \tag{16}$$

It follows from the orthogonality condition that

$$Q_n^* Q p^n(H) Q^* b = 0 = Q_n^* [Q_n U] p^n(H) Q^* b = [p^n(H), 0] e_1 \|b\| = 0. \tag{17}$$

This amounts to the condition that the first n entries of $p^n(H)$ are zeros. Because of the structure of H, $p^n(H_n)$ has the same structure. By the Cayley-Hamilton, theory, we can choose the characteristic polynomial of H_n as one candidate polynomial. We now prove the uniqueness. Suppose there is another polynomial p^n which satisfies $p^n(A)b \perp K_n$. Take the difference would give a nonzero polynomial q of order $n-1$ with $q(A)b = 0$. This contradicts with the assumption that K_n is of full rank.

□