1 April 3rd, 2019

1.1 Shifted Inverse Power Iteration Part 2

Let $\lambda_i,\ i = 1, 2, \ldots, n$ with $|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|$ be eigenvalues of A. Note that $(\lambda_i - \mu)^{-1}$ are eigenvalues of $(A - \mu I)^{-1}$.

Thus we can apply the power iteration to $(A - \mu I)^{-1}$:

Algorithm 1.1
Choose $y^{(0)} \in \mathbb{R}^n$ s.t. $\|y^{(0)}\|_2 = 1$
For $k = 1, 2, \ldots$

- $z^{(k)} = (A - \mu I)^{-1} y^{(k-1)}$ (Done by solving $(A - \mu I)z^{(k)} = y^{(k-1)}$.
- $y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2}$.
- $\mu^{(k)} = (y^{(k)})^T A y^{(k)}$ (the Rayleigh Quotient of A)

1. This iteration is the "shifted inverse power iteration"
2. To make the iteration converge to (λ_j, x_j), the following has to be satisfied:
   (a) μ is chosen s.t. $\frac{1}{|\lambda_j - \mu|}$ is the largest among $\frac{1}{|\lambda_i - \mu|}$

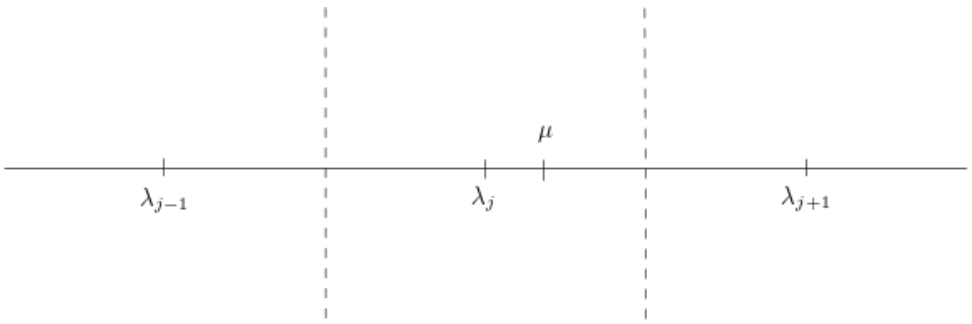

Figure 1: Choosing μ to guarantee convergence

   (b) $\langle y^{(0)}, x_j \rangle \neq 0$. (or else it will converge to another eigenvector.
   (c) The convergence rate depends on:

$$\frac{\frac{1}{|\lambda_j' - \mu|}}{\frac{1}{|\lambda_j - \mu|}} = \frac{|\lambda_j - \mu|}{|\lambda_j' - \mu}$$

   , where $\frac{1}{|\lambda_j' - \mu|}$ is the second largest among $\frac{1}{|\lambda_i - \mu|}$ and

$$|\lambda_j - \mu| < |\lambda_j' - \mu|.$$

   (d) For an ϵ- precision solution, the computational complexity is

$$O(n^3) + O\left(\log \frac{1}{\epsilon} \cdot n^2\right),$$

   since we only compute the LU decomposition of $(A - \mu I)$ once.

To accelerate the shifted power iteration, we can also use an adaptive shift (if we shift μ to be closer to the target eigenpair, it will converge faster).

Algorithm 1.2 (Rayleigh Quotient Iteration)
Choose $y^{(0)} \in \mathbb{R}^n$ s.t. $\|y^{(0)}\|_2 = 1$
$\mu^{(0)} = (y^{(0)})^T A y^{(0)}$
For $k = 1, 2, \ldots$

- $z^{(k)} = (A - \mu^{(k-1)} I)^{-1} y^{(k-1)}$ (Done by solving $(A - \mu I) z^{(k)} = y^{(k-1)}$.
- $y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2}$.
- $\mu^{(k)} = (y^{(k)})^T A y^{(k)}$ (the Rayleigh Quotient of A)

i.e. we choose μ to be close to the desired eigenvalue.

- This converges to some eigenpair (λ_i, x_i) such that λ_i is close to $\mu^{(0)}$.
- This Rayleigh Quotient iteration converges very fast (cubic).
- However since $(A - \mu^{(k-1)} I)$ is changing, we have to calcualte the LU decomposition each time.
- However, this will only converge to one eivenpair.

1.2 Simultaneous Iteration

To compute r eigenpairs:

Algorithm 1.3
Choose $Y^{(0)} \in R^{n \times r}$ s.t. $(Y^{(0)})^T (Y^{(0)}) = I$
For $k = 1, 2, \ldots$

- $Z^{(k)} = A Y^{(k-1}$
- Set $Y^{(k)}$ to be the Q matrix in the QR decomposition of $Z^{(k)}$.
- $\mu_i^{(k)} = (y_i^{(k)})^T A y_i^{(k)}$, where $y_i^{(k)}$ is the i-th column of $Y^{(k)}$

Under some assumption, we have:

$$\|y_i^{(k)} - \pm x\| \leq C \rho^k, i = 1, \ldots, r,$$

where $\rho = \max_{i=1,\ldots,r} \frac{|\lambda_i + 1}{\lambda_i} < 1$ and $|\mu_i^{(k)} - \lambda_i| \leq C \rho^k$

1.3 QR algorithm for Eigenvalue Decomposition

We set $r = n$ in the simultaneous power iteration

$$Z^{(k)} = A Y^{(k-1)}$$

$$Y^{(k)}R^{(k)} = Z^{(k)},$$

i.e. let $Z^{(k)} = Y^{(k)}R^{(k)}$ be the QR decomposition of $Z^{(k)}$. Eliminating $Z^{(k)}$, we have

$$Y^{(k)}R^{(k)} = AY^{(k-1)} \iff (Y^{(k)})^T AY^{(k-1)} = R,$$

because $(Y^{(k)})^T Y^{(k)} = Y^{(k)}(Y^{(k)})^T = I$.

Let $A^{(k)} = (Y^{(k)})^T AY^{(k)}$, then

$$A^{(k-1)} = (Y^{(k-1)})^T AY^{(k-1)} = (Y^{(k-1)})^T Y^{(k)} R^{(k)}.$$

Since $(Y^{(k-1)})^T Y^{(k)}$ is orthogonal and $R^{(k)}$ is upper triangular, $A^{(k)}$ is just an orthogonal square matrix. Note that

$$A^{(k)} = (Y^{(k)})^T AY^{(k)} = (Y^{(k)})^T AY^{(k-1)}(Y^{(k-1)})^T Y^{(k)} = R^{(k)}(Y^{(k-1)})^T Y^{(k)}.$$

This means that after getting the QR decomposition of $A^{(k-1)}$, we swap the two matrices to get $A^{(k)}$.

Algorithm 1.4 (QR Algorithm)
Choose initial guess $A^{(0)} = (Q^{(0)})^T \forall Q^{(0)}$, e.g. $Q^{(0)} = I \implies A^{(0)} = A$.
For $k = 1, 2, \ldots$

- Compute the QR decomposition: $A^{(k-1)} = Q^{(k)}R^{(k)}$.
- Set $A^{(k)} = R^{(k)}Q^{(k)}$.

Remark 1.5 —

1. Note that $A^{(k)} = R^{(k)}Q^{(k)} = (Q^{(k)})^T A^{(k-1)}Q^{(k)}$. By induction

   $$A^{(k)} = (Q^{(k)})^T \cdots (Q^{(1)})^T (A^{(0)})^T AQ^{(0)}Q^{(1)} \cdots Q^{(k)} \implies A^{(k)} \text{ is similar to } A$$

   as $Q^{(0)}Q^{(1)} \cdots$ is an orthogonal matrix.

2. Since $Y^{(k)}$ is expected to converge to the eigenvectors of A,

   $$A^{(k)} = (Y^{(k)})^T AY^{(k)} \text{ is expected to converge to } \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}.$$

   It can be proven that if the eigenvalues of A are well separated, then $A^{(k)} \to \Lambda$ and $Q^{(0)} \cdots Q^{(k)}$ converges to the eigenvectors of A.

3. Since QR decomposition and matrix-matrix product costs $O(n^3)$, the total computational cost is

   $$O(kn^3), \text{ where } k \text{ is the number of iteration needed.}$$

   Note that if k iterations is done (or if $k \sim O(n)$), then it will be $O(n^4)$, which is expensive.

Note that even though the QR decomposition is not unique, it will still work since the QR have similar properties/structure.

1.4 Practical Implementation of QR Algorithm

The idea is to choose $Q^{(0)}$ such that $A^{(0)}$ is "well structured". This will allow QR decomposition to be done in $O(n^2)$. For our purpose, this "structure" is to be tridiagonal.

Thus this implementation has two phases:

1. Find $Q^{(0)}$ such that $A^{(0)} = (Q^{(0)})^T A Q^{(0)}$ is tridiagonal.
2. QR decomposition of tridiagonal matrices is done in $O(n^2)$ and so that $A^{(k)}$ preserves the tridiagonal structure.

Algorithm 1.6 (Phase 1)
We will use the Householder transformation.

1. Let P_1 be the Householder transform s.t.

$$P_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & H_1 & \\ 0 & & & \end{bmatrix} : \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \\ \times & \times & & \times \end{bmatrix} \to \begin{bmatrix} \Delta & \Delta & \cdots & \Delta \\ \times & \times & \cdots & \times \\ 0 & & & \\ \vdots & \vdots & \ddots & \\ 0 & \times & & \times \end{bmatrix}.$$

Then $P_1 A P_1^T$