QR decomposition for large scale matrices

Andrew Feng Department of Computer Science and Technology

To diagonalize a matrix, it is always important to get the eigenvalues of one matrix, but to get the eigenvalues of large matrices is not always easy, which is also fairly important in Computer Science Numerical Analysis algorithm designing.

When we are going to compose the eigenvalues of a matrix A, usually we will perform the characteristic polynomial of this matrix $p_A(x) = det(\lambda I - A)$, the roots of which shall be all eigenvalues of it while the exponents of each term $(\lambda_i - x)$ is the algebric counting multiciplity of this eigenvalue. But this is not always the case. This algorithm is involved with a matrix determinant composing along with a polynomial equation roots composing. In most cases we encountered, this might be a good algorithm since these two are easy to tackle with when A is no more then 4 to 5 dimensions. But suppose if we get a 'large' matrix, say 10 by 10, it is not always so pleasing to compose $\lambda I - A$ while get the roots of a 10-order polymonial equation. ***Example 1.***(a simple case in our usual tests)

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & -2 \\ 1 & -1 & 2 & -2 \\ -2 & 0 & 0 & 3 \end{bmatrix}$$
$$then\ det(\lambda I - A) = \begin{bmatrix} \lambda - 1 & 0 & 0 & 0 \\ -2 & \lambda - 1 & 0 & 2 \\ -1 & 1 & \lambda - 2 & 2 \\ 2 & 0 & 0 & \lambda - 3 \end{bmatrix}$$
$$and\ the\ characteristic\ polymonial\ of\ this\ A\ is\ p_A(x) = (\lambda - 1)^2(\lambda - 2)(\lambda - 3)\ ,\ so\ the\ eigenvalues\ are\ obviously\ 1,\ 1,\ 2\ and\ 3.$$

Example 2.(a tough case)

$$A = \begin{bmatrix} 2 & 2 & 2 & 1 & 0 & 7 & 0 & 1 \\ -4 & 6 & -2 & 2 & 0 & -4 & 5 & -3 \\ 5 & -18 & 8 & 0 & 6 & 1 & 4 & 2 \\ 2 & 8 & 0 & 6 & 0 & 3 & 5 & -2 \\ 7 & -10 & 8 & 6 & 5 & 9 & 6 & 0 \\ 3 & -4 & 6 & 8 & 5 & 4 & 12 & -1 \\ -1 & 2 & 4 & 10 & 5 & 0 & 21 & -4 \\ 1 & 4 & 6 & 11 & 5 & 7 & 21 & -2 \end{bmatrix}$$

It is kind of scary to compose the eigenvalues of such a matrix. The computing of $det(\lambda I - A)$ is too awful. But from determinant to $p_A(x)$ in beyond form is also non-trival.

In our daily life, the matrices generated from real models like medical CTs own generally distinct eigenvalues due to the randomality. In this way for any n-square matrices A in this matter, it can always follows the spectrum theory, i.e. there is a primary and invertible matrix P, such that $A = PDP^{-1}$, while D is diagonal matrix with all eigenvalues of A on it diagonal line. In this term we've learned that not all matrices follows the spectrum theory condition, i.e, they are not always diagonalizable. But according to Jordan Normal Form theory, they are always similar to a quasi-upper triangular matrix, the Jordan Normal Form. Therefore, for any matrix, it is always similar to a diagonal or quasi-diagonal matrix, the eigenvalues of which lay in the diagonal line. However, to get the eigenvalues, one still has to compose $det(\lambda I - A)$.

Nevertheless, if we lower the standard for eigenvalues, say if we just have to get the approximate values of them, there are better and easier solutions. The easiest way to do this is the Getschgorin circle theorem, which can give us a circular area of the distribution of an eigenvalue on a 2-dim complex coordinate plane.

Now consider from another perspective. We have Newton-Raphson method to solve a equation by applying a matrix or function over and over again to the solution and finally the solution converges to a staitc point, which is the answer. If we can use similar method to get the eigenvalues of A by applying something over and over again to A, this might be easier for us to solve the $det(\lambda I - A)$. Since in this term we've learned Schur's Theorem, luckily we got a way through QR decomposition.

Theorem 1 . (Schur's Decomposition Theorem)

if A is a $n \times n$ square matrix with complex entries, then A can be expressed as

$$A = QUQ^{-1}$$

where Q is a unitary matrix and U is an upper triangular matirx.[1]

How effective is this technique? If we have to do QR decomposition for numerous times, this might be disgusting(but still you cannot compute the $det(\lambda I - A)$ in Example 2, and to get the eigenvalues we have to try....).

There must be a ruler to measure the convergence speed of it and there are techniques to accelerate this procedures. Let's see some examples.

Example 3. (diagonalized or upper/lower triangular matrices' eigenvalues can be read through their diagonal line)

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

This A is a Jordan Block, we can find its eigenvalues in two perspectives. First directly find all the eigenvalues in the diagonal line, since it is definition. On the other hand, the QR decomposition of A is $Q = I, R = A$, and it have already converged.

Example 4. (Quasi-Upper/Lower triangular matrices)

Though not perfect, we still want a 'good' form of such matrices that we can get a QR decomposition at a relatively low cost. Say if the matrix is Upper Triangular, we add a vice-diagonal line ajacent to the diangonal line.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 8 \end{bmatrix}$$

It is noteworthy that these kind of matrices have convenient way of QR decomposition, the complexity of which must be O(n). This confidence is *given* by ***Givens*** (name of the inventer) and it is called ***Givens Transformation***. Look at the first column, we can find two non-zero elements and we can perform a circular transformation: $(1, 1) \to (\sqrt{2}, 0)$, and this is orthogonal. This is an operation between first and second row. For all $a_{i,i-1} \neq 0$, we can perform this row by row and thus got a orthogonal matrix Q, such that A = QR. (For more information, refer to Givens Transformation).

Still, do to Schur's theorem, A is similar to a quasi-Triangular matrix. So we aplly RQ and give its output to A. This process can be done by computers (yet the $det(\lambda I - A)$ does not seem to be so trival for machines).

The result of first epoch:

$$Q = \begin{bmatrix} -0.707 & 0 & 0.236 & -0.533 & 0.358 & -0.176 & 0.031 & 0.011 \\ -0.707 & 0 & -0.236 & 0.533 & -0.358 & 0.176 & -0.031 & -0.011 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.943 & -0.267 & 0.179 & -0.088 & 0.015 & 0.005 \\ 0 & 0 & 0 & -0.6 & -0.716 & 0.352 & -0.062 & -0.022 \\ 0 & 0 & 0 & 0 & -0.447 & -0.879 & 0.155 & 0.054 \\ 0 & 0 & 0 & 0 & 0 & -0.183 & -0.928 & -0.324 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.329 & 0.944 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 2 & -0.707 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.414 & 1 & -0.471 & 1.067 & -0.716 & 0.352 & -0.062 & -0.022 \\ 0 & -2.121 & 3.556 & 1.006 & -0.675 & 0.332 & -0.058 & -0.02 \\ 0 & 0 & 1.571 & 2.244 & 1.848 & -0.909 & 0.16 & 0.056 \\ 0 & 0 & 0 & 2.683 & 4.4 & 0.868 & 0.277 & 0.097 \\ 0 & 0 & 0 & 0 & 2.441 & 5.162 & 0.992 & 0.346 \\ 0 & 0 & 0 & 0 & 0 & 1.113 & 6.505 & -0.521 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2.488 & 7.132 \end{bmatrix}$$

The theoretical accurate eigenvalues of A are [8, 7, 5, 5, 4, 2.732, 1, -0.732], if we tolerate $\pm 10\%$ as errors, we repeat the procedure for 10 times and we get

$$A_{10} = \begin{bmatrix} 7.088 & 0.116 & 0.412 & 0.16 & 1.07 & 0.513 & 0.158 & 0.2 \\ -1.273 & 6.432 & -0.99 & -0.098 & 1.046 & 0.298 & 0.371 & 0.466 \\ 0 & -2.133 & 6.682 & 0.106 & 0.263 & 0.641 & 0.41 & 0.502 \\ 0 & 0 & -1.466 & 4.764 & -0.142 & 0.661 & 0.486 & 0.557 \\ 0 & 0 & 0 & 0.924 & 4.23 & -0.177 & -0.222 & -0.121 \\ 0 & 0 & 0 & 0 & 0.487 & 2.536 & -0.431 & -1.113 \\ 0 & 0 & 0 & 0 & 0 & 0.001 & 0.987 & 0.052 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.406 & -0.72 \end{bmatrix}$$

As we can see, the low sub-diagonal line's elements generally shrinks to 0 and the diagonal line elements generally limit to eigenvalues. Except for 5 (with 2 algebriccounting multiplicities, at this time they correspond to 6.682 and 4.764 repectively), other element reached the given accuracy. This is the prediction of Schur's Theorem that such procedure will finally converges at a upper triangular matrix (if we properly arrage the position of eigenvalues via applying permutation matrix P and P^{-1} on both sides of A, we will get a block diagonal matrix with each block less than 2 rank).

Techniquely, this kind of matrices is called a Upper Hessenburg Matices.

Here comes the codes for running such program for numpy (python3, anaconda), which uses widely the same package as Matlab

```
import numpy as np

np.set_printoptions(suppress=True)
np.set_printoptions(precision=3)

A = np.array([[1, 1, 0, 0, 0, 0, 0, 0],
              [1, 1, 1, 0, 0, 0, 0, 0],
              [0, 2, 1, 0, 0, 0, 0, 0],
              [0, 0, 2, 4, 0, 0, 0, 0],
              [0, 0, 0, 1, 5, 0, 0, 0],
              [0, 0, 0, 0, 2, 6, 1, 0],
              [0, 0, 0, 0, 0, 1, 6, 0],
              [0, 0, 0, 0, 0, 0, 2, 8]])
q, r = np.linalg.qr(A)
print(q)
print(r)
s = np.matmul(r, q)
print(s)

for i in range(1, 10):
    q, r = np.linalg.qr(s)
    print(q)
    print(r)
    s = np.matmul(r, q)
    print(s)

print(np.linalg.eigvals(A))
```

Example 5. (symmetric matrices)

$$A = \begin{bmatrix} 1 & 3 & 4 \\ 3 & 1 & 2 \\ 4 & 2 & 1 \end{bmatrix}$$

For the convenience of our discussion, this is a 3-order matrix. One can compute its eigenvalues in $det(\lambda I - A)$ but this is not what I want to show.

I claim that first I can apply a Householder transformation: for the low triangular part of this matrix, we can get

$$H_1 = I - 2\vec{w}\vec{w}^T = \begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix}$$
$$H = \begin{bmatrix} 1 & 0 \\ 0 & H_1 \end{bmatrix}$$
$$HessenburgH = HAH^T = \begin{bmatrix} 1 & -5 & 0 \\ -5 & 2.92 & 0.56 \\ 0 & 0.56 & -0.92 \end{bmatrix}$$

Here we get the Hessenburg H and then we can compose the algorithm given in example 4 on H can get a approximate values of A.

The accurate eigenvalues of A are 7.075, -3.188, -0.887. Running former procedure for 3 times, we got approximate eigenvalues of A from the diagonal line of A_3

$$A_3 = \begin{bmatrix} 6.954 & 1.107 & 0 \\ 1.107 & -3.067 & -0.008 \\ 0 & -0.008 & -0.887 \end{bmatrix}$$

and this precision is generally tolerable.

Notice that in this case A is a symmetric matrix, for symmetric matrices, when we apply $Q^T AQ$ on A, this is a congruence transformation, and $Q^T AQ$ is also symmetric. For computers, when tacking symmetric matrices, they can just store the lower triangular part of this matrix. When we perform QR decomposition on A, $Q^T A$ affect the upper part of A while QR affect only lower part of A, so only half of the matrix multiple has to be calculated, making the problem simpler.

Example 6. (convergence speed)

consider a diagonalizable matrix A and a non-diagonalizable matrix B, such that

$p_A(x) = (\lambda - 1)^2(\lambda - 2)(\lambda - 3) = p_B(x) = m_A(x),\ m_B(x) = (\lambda - 1)(\lambda - 2)(\lambda - 3)$,$m_A(x), m_B(x)$ are the minimal polymonials of A and B.

Without loss of generosity, A has similar normal form

$$D = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 2 & \\ & & & 3 \end{bmatrix}$$

B has Jordan Normal Form

$$J = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

Let

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & -2 \\ 1 & -1 & 2 & -2 \\ -2 & 0 & 0 & 3 \end{bmatrix} B = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 2 & 0 & 0 & -3 \\ 1 & -1 & 2 & 1 \\ -2 & 1 & 0 & 4 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 \\ 0 & 1 & 1 & -1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \; s.t. \; A = P^{-1}DP, \; B = P^{-1}JP$$

that is to say, A and B come from exactly the same operation from its normal form, the only difference of which is whether diagonalizable or not.

We set the tolerence for errors is 20% per eigenvalue, and we perform the former procedure to test the quickest time of reaching such status. We found that a non-diagonalizable B tool 5 times to reach, while diagonalizable A took 3 times to reach. Blow are results and codes.

$$n = 3$$

$$A_3 = \begin{bmatrix} 3.148 & 0.286 & -2.875 & -2.837 \\ -0.14 & 2.019 & 0.846 & -0.425 \\ 0.017 & -0.11 & 0.907 & 0.044 \\ 0.05 & 0.02 & -0.059 & 0.926 \end{bmatrix}$$

$$n = 5$$

$$A_5 = \begin{bmatrix} 3.048 & 0.571 & -2.84 & -2.918 \\ -0.064 & 1.98 & 0.728 & -0.634 \\ 0.002 & -0.026 & 0.981 & 0.017 \\ 0.005 & 0.003 & -0.006 & 0.992 \end{bmatrix}$$

$$B_5 = \begin{bmatrix} 3.329 & -1.321 & -0.71 & 3.039 \\ 0.227 & 2.14 & 2.104 & -2.834 \\ -0.035 & -0.168 & 0.828 & 1.907 \\ -0.001 & -0.004 & -0.026 & 0.803 \end{bmatrix}$$

$$define\ a\ new\ norm : F(A) = \sum_{i=1}^{n} |a_{ii}|$$

$$F(A_3 - diag(3,2,1,1)) = 0.148 + 0.019 + 0.093 + 0.074 = 0.334$$
$$F(A_5 - diag(3,2,1,1)) = 0.048 + 0.020 + 0.019 + 0.008 = 0.095$$
$$F(B_5 - diag(3,2,1,1)) = 0.329 + 0.140 + 0.172 + 0.197 = 2.100$$

```python
import numpy as np

np.set_printoptions(suppress=True)
np.set_printoptions(precision=3)

B = np.array([[1, 1, 0, 1],
              [2, 0, 0, -3],
              [1, -1, 2, 1],
              [-2, 1, 0, 4]])
q, r = np.linalg.qr(B)
print(q)
print(r)
s = np.matmul(r, q)
print(s)

for i in range(1, 5):
    q, r = np.linalg.qr(s)
    print(q)
    print(r)
    s = np.matmul(r, q)
    print(s)

print(np.linalg.eigvals(B))
```

from the results above, we can see that a diagonalizable matrix A converges far faster and more accurate than a non-diagonalizable matrix B with same eigenvalues. This reflects that this QR decomposition is influenced by the 'failure' of spectrum theorem.

More examples can ge generated in this way.

Example 7:

$$A = \begin{bmatrix} 1 & 4 & -2 & 0 & -4 & 0 \\ 4 & -13 & 8 & 0 & 12 & 0 \\ 8 & -30 & 18 & 0 & 26 & 0 \\ -9 & -33 & -18 & 2 & -27 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 15 & -90 & 42 & 0 & 78 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 4 & -2 & 0 & -4 & 0 \\ 6 & -52 & 20 & 0 & 51 & 0 \\ 16 & -126 & 50 & 0 & 122 & 0 \\ -15 & 129 & -48 & 2 & -123 & 0 \\ -2 & 7 & -4 & 0 & -6 & 0 \\ 27 & -342 & 120 & 0 & 320 & 4 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$J = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ -3 & -3 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 5 & -4 & 0 & 0 & 1 \end{bmatrix}$$

The times to reach the convergence of A and B are 3 and 10 respectively. Here we suppose the errors between the theoretical and approximate value is no more than 20% for each eigenvalue (for 10%, they are 10 and 12 respectively and we can find that the failure of spetrum influence the initial convergence speed much more) . We got the same conclusion with Example 6.

Proposition(might be false): If two matrices have exactly same eigenvalues(with same algebirc counting multiplicity), QR decomposition method for finding approximate eigenvalues converges faster for the matrix that has larger minimal polymonial degree, i.e, if deg($p_A(x)$)-deg($m_A(x)$) < deg($p_B(x)$)-deg($m_B(x)$), then A converges faster than B.

Here, I searched for several materials for numerical analysis but found no standard for measuring this speed. An attemping idea is that if we define a special norm to measure the eigenvalues of a matrix, life the 'F-norm'(not Frobenius, I invented this one myself), we can see that error for ajacent two operations might be related, if $\lim_{n->+\infty} \frac{F(A_{n+1}-D)}{F(A_n-D)^p}$ is somewhat constant c, p is an positive number greater than 1, than this p might be a measure for the failure of spectrum theorem. Well, there might be some correlation between p and $det(\frac{A^HA-AA^H}{2})$, where A^H is taking transpose conjugate.

Summary: In our real life, the matrices are not always form to be 'good' numbers, and in this way we do not necessarily have to solve the exact eigenvalues of a matrix. Apart from Getschgorin Circle Theorem, QR decomposition, based on Schur' Theorem, provide a more accurate and flexible approximation of the eigenvalues of a matrix. The convergence speed has something to do with diagonalizability, the less failure a matrix is from spectrum theorem, the faster it converges.

Reference

[1]Schur Decomposition, Wikipedia. https://en.wikipedia.org/wiki/Schur_decomposition.

Note

The QR decomposition used in the text is the package by numpy linalg qr, as far as I know, it utilize Householder transformation along with Givens transformation to achieve it. So there are no obivious 'theoretical' circle - proving two things rely on each other.