10-12-2023

Numerical Analysis

Eigenvalues and eigenvectors

Haka Kevin

Introduce

Linear algebra is one of the most important branches of mathematics. It has applications in almost every field of engineering and natural science. Many problems that scientists and engineers are called to solve belong in the field of linear algebra. As we know, linear algebra deals with the simultaneous solving of many linear equations. It's not hard to understand that if we have more than a few linear equations to solve, the problem becomes very hard to solve by hand. If the number of linear equations overcome three digits, which is very common for real problems, the system becomes hard even for computers to solve it efficiently in a short time. That's why we need algorithms to solve such problems with the minimum number of operations, like Gauss elimination and LU decomposition. As for the power method, it is a very easy method that doesn't need to solve the system of linear equations to find the dominant eigenvalue and its corresponding eigenvector. There are many problems where finding the eigenvalues and the eigenvectors of a system is important. For example, in quantum mechanics, the energy of the ground state of the electron in the atom of hydrogen is nothing more than the dominant eigenvalue of the Schrödinger equation for the electron in the atom of hydrogen, and the corresponding eigenvector is the wave equation at the ground state of the system. That's why these methods are very important, they help us solve problems in an approximate but consistent way, so that the results can be compared with theoretical models or experimental results in any natural science.

In this work, we are going to solve the matrix A below by finding its inverse matrix with LU decomposition. Then, we will approximate the dominant eigenvalue and its corresponding eigenvector using the power method, which is a very fast method that doesn't require solving the characteristic polynomial.

$$A = \begin{bmatrix} 4 & 1 & 2 & 3 & 5 \\ 1 & 3 & 1 & 4 & 2 \\ 2 & 1 & 5 & 2 & 3 \\ 3 & 4 & 2 & 4 & 1 \\ 5 & 2 & 3 & 1 & 5 \end{bmatrix}$$

LU Decomposition

Theory

The main idea of LU decomposition is very simple. If we break a matrix A which is the matrix of coefficients in a linear equation $Ax = b$, into two other matrices L and U (A = LU), where L is a lower triangular matrix and U is an upper triangular matrix, we can use backward and forward substitution to find the **x** vector or the inverse matrix of A. With L and U, the calculation of the determinant of A is also very straightforward.

Let's start with

$$Ax = b \tag{1}$$

If $A = LU$ we have

$$LUx = b \tag{2}$$

We define

$$Ux = c \tag{3}$$

$$Lc = b \tag{4}$$

If we knew L and U matrices, we could use equation (4) with forward substitution to determine the **c** vector. With vector **c** and equation (3), we could do backward substitution to determine the vector **x**, which is the solution of our system. The method is easy to implement, but we have one problem, we need to determine the matrices L and U.

The way to determine the U matrix is by using Gauss elimination on A, and the elements of matrix L will be the multipliers f that were used to make the respectively elements of A zero.

There are many pairs of L and U matrices whose multiplication (LU) can give the A matrix. Two of the most common implementations are Doolittle decomposition, which defines the L matrix with 1's on the main diagonal, and Crout decomposition, which defines the U matrix with 1's on the main diagonal (Steven C. Chapra, et al., 2015). In the algorithm implementation below, Crout decomposition is used.

Crout decomposition

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{11} & \dots & l_{nn} \end{bmatrix} \qquad U = \begin{bmatrix} 1 & u_{12} & \dots & u_{1n} \\ 0 & 1 & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

The algorithm implementation of the Crout decomposition is based on the fact that we can determine the elements of L and U matrices gradually, as shown in Figure 1. The elements of the first column of L are equal to the elements of the first column of A, the elements of the first row of U are equal to some operations between the elements of the first row of A and the elements of the first column of L, the elements of the second column of L are equal to some operations between known elements of L and U and the elements of the second column of A, and so on.

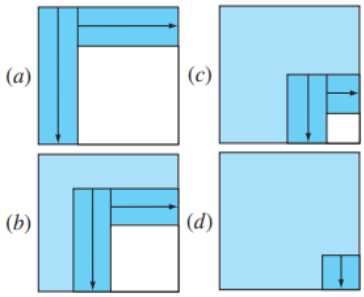

Figure 1 Gradually determination of L and U matrices elements.

The algorithm

$$l_{i1} = a_{i1} \; for \;\; i = 1,2, \dots, n \tag{5}$$

$$l_{1,j} = \frac{a_{1j}}{l_{11}} \; for \;\; j = 2,3, \dots, n \tag{6}$$

$$For \;\; j = 2, 3, \dots, n-1$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} \, u_{kj} \; for \;\; i = j, j+1, \dots, n \tag{7}$$

$$u_{jk} = \frac{a_{jk} \; - \sum_{i=1}^{j-1} l_{ji} \, u_{ik}}{l_{ii}} \; for \;\; k = j+1, j+2, \dots, n \tag{8}$$

$$l_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} \, u_{kn} \tag{9}$$

The LU decomposition is a very elegant and efficient method to determine the inverse matrix of A. To do that, if we know the L and U matrices, we can use equation 11, where $\mathbf{I_i}$ represents the ith column of the identity matrix and $\mathbf{x_i}$ represents the ith column of the inverse matrix of A. The inverse matrix can be calculated column by column with a backward and a forward substitution for each of them.

$$Ax_i = I_i \tag{10}$$

$$LUx_i = I_i \tag{11}$$

Results

If we use the above algorithm for our matrix A, we obtain the following matrices L and U where the elements are rounded to three decimal places for better readability. However, the calculations have been performed with greater precision.

$$L = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 1 & 2.75 & 0 & 0 & 0 \\ 2 & 0.5 & 3.909 & 0 & 0 \\ 3 & 3.25 & -0.091 & -2.093 & 0 \\ 5 & 0.75 & 0.364 & -3.628 & 4.8 \end{bmatrix} \quad U = \begin{bmatrix} 1 & 0.25 & 0.5 & 0.75 & 1.25 \\ 0 & 1 & 0.181 & 1.182 & 0.273 \\ 0 & 0 & 1 & -0.023 & 0.093 \\ 0 & 0 & 0 & 1 & 1.733 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

By definition, we know that $A = LU \Rightarrow det(A) = det(L)det(U)$. The determinant of a triangular matrix is just the multiplication of its main diagonal elements. In our case, none of the main diagonal elements of L are zero, which leads us to $det(L)det(U) = -432 \Rightarrow det(A) \neq 0$. This means that our matrix A has an inverse. With equation 11 we calculate the inverse of A.

$$A^{-1} = \begin{bmatrix} 0.245 & -0.458 & -0.102 & 0.343 & -0.069 \\ -0.458 & 0.375 & -0.083 & -0.083 & 0.375 \\ -0.102 & -0.083 & 0.259 & 0.037 & -0.028 \\ 0.343 & -0.083 & 0.037 & 0.148 & -0.361 \\ -0.069 & 0.375 & -0.028 & -0.361 & 0.208 \end{bmatrix}$$

Power method

Theory

The most common way to determine the eigenvalues and eigenvectors of a matrix is by using its characteristic polynomial. However, the roots of a polynomial are very sensitive to its coefficients. As we know, computational methods always have an error which is controllable but not zero. For big matrices, this error can accumulate and guide us to extremely wrong roots for the characteristic polynomial.

The power method is an iterative method which approximate in each iteration the dominant eigenvalue of the matrix A. Let's suppose that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. It can be proved that we can approximate the dominant eigenvalue λ_1 with the equation 12. The x_{n+1} is defined at the equation 13, by definition x_0 is a non-zero random vector. The reason we start with a random vector, is to tend the chance to zero that our vector is orthogonal to the eigenvector associated with the dominant eigenvalue.

The rate of convergence in this method is depends on ratio $\left|\frac{\lambda_2}{\lambda_1}\right|$. If this ratio is close to 0, the method converges very fast. If the ratio is close to 1, which means $\lambda_1 \approx \lambda_2$, the method converges very slowly, and if the ratio is equal to 1, the method does not converge (Demmel, 1997).

To stop the iteration in this method, we can use the Scarborough criterion (equation 14) in comparison with the relative error of the eigenvalue (equation 15), which assures us that if the criterion $\boldsymbol{\varepsilon_\lambda < \varepsilon_s}$ is true, we have at least **'n'** significant figures.

$$\lambda_1^{new} = \frac{x_n^T \cdot x_{n+1}}{x_n^T \cdot x_n} \tag{12}$$

$$x_{n+1} = A \cdot x_n \tag{13}$$

$$\varepsilon_s = \frac{1}{2} 10^{-n} \tag{14}$$

$$\varepsilon_\lambda = \left| \frac{\lambda_1^{new} - \lambda_1^{old}}{\lambda_1^{new}} \right| \tag{15}$$

The algorithm

1. Produce a random vector $\mathbf{x_{old}}$.
2. Culculate $\mathbf{x_{new}}$ (equation 13).
3. Culculate $\boldsymbol{\lambda_{new}}$ (equation 12).
4. set $\boldsymbol{\lambda_{old} = \lambda_{new}}$ and $\boldsymbol{x_{old} = \frac{x_{new}}{(x_{new}^T \cdot x_{new})^{1/2}}}$.

In a loop:

5. Culculate $\mathbf{x_{new}}$.
6. Culculate $\boldsymbol{\lambda_{new}}$.
7. Culculate $\boldsymbol{\varepsilon_\lambda}$.
8. Control if $\boldsymbol{\varepsilon_\lambda < \varepsilon_s}$.
9. If step **8** is True stop.
10. Else
    set $\boldsymbol{\lambda_{old} = \lambda_{new}}$ and $\boldsymbol{x_{old} = \frac{x_{new}}{(x_{new}^T \cdot x_{new})^{1/2}}}$
    and go step **5**.

Warning: In step **5** and **10**, the array that is set in $\mathbf{x_{old}}$ is normalized to avoid overflow.

Results

Based on the algorithm above and the condition to find the dominant eigenvalue with 4 significant figures, we obtained Table 1 as a result. The eigenvectors are written with three decimal places for greater clarity, but the calculations have been performed with greater precision. Finally, it is important to note that the power method is a stochastic process, and the iteration number and the results from the method will change slightly every time the algorithm runs.

Table 1 The results from the power method, founding the eigenvalue with 4 significant figures and its eigenvector.

Iteration	λ	v	ε_λ	ε_s
1	10.252723	[0.446, 0.276, 0.463, 0.45, 0.555]	NaN	0.00005
2	13.935328	[0.508, 0.332, 0.432, 0.408, 0.528]	0.264264	0.00005
3	14.070944	[0.504, 0.329, 0.42, 0.417, 0.536]	0.009638	0.00005
4	14.075419	[0.506, 0.33, 0.418, 0.417, 0.535]	0.000318	0.00005
5	14.075603	[0.506, 0.331, 0.417, 0.418, 0.535]	0.000013	0.00005

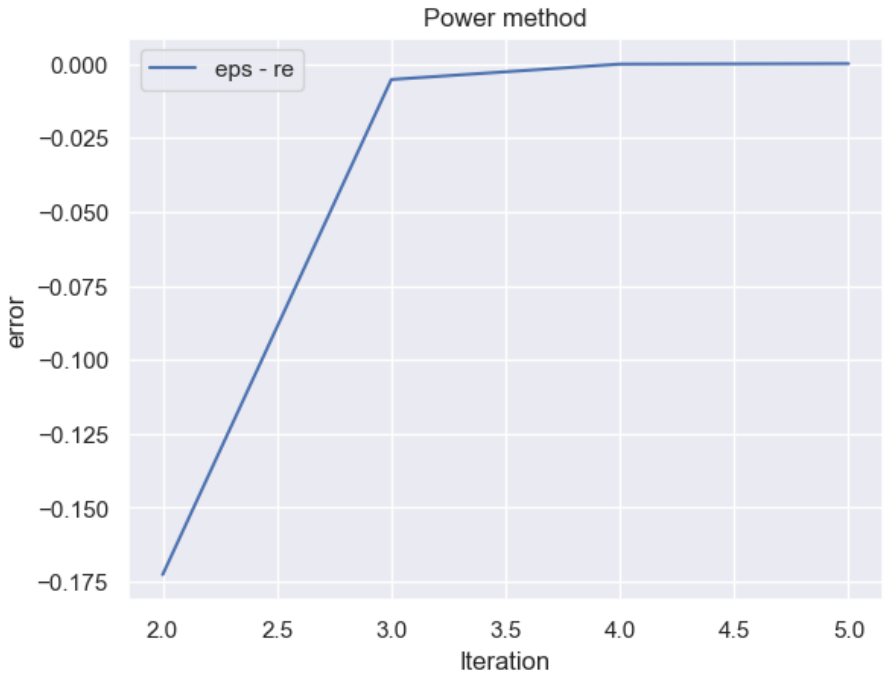

Figure 2 In the plot, we can see how the method converges in each iteration. The graph shows the Scarborough criterion - relative error in each iteration.

Conclusion

As we showed above, the methods are very easy to implement and code. Generally speaking, they seem to be fast methods with good results for both of them. The LU decomposition has an $O(n^3)$ complexity which is better than many methods that solve a linear problem. It has the same complexity as the Gauss elimination method, but the big advantage of the LU decomposition is that for another **b** vector, the complexity reduces to $O(n^2)$. This is because the L and U matrices are independent of the **b** vector and don't need to be recalculated, which is the time-consuming step. As for the power method, in our case, it was very fast. However, as we mentioned above, it depends on the ratio of the two first dominant eigenvalues. There is an extended version of the power method known as the inverse power method, which can find the other eigenvalues except the dominant one. However, the complexity, as it is reasonable, increases.

References

Demmel, James W. 1997. Applied Numerical Linear Algebra. s.l. : SIAM, 1997.

Steven C. Chapra and Raymond P. Canale. 2015. Numerical Methods. New York : McGraw-Hill Education, 2015.