XIAMEN UNIVERSITY MALAYSIA
厦門大學馬來西亞分校

MAT103 Project Report

Analysis of Power Sequence of Power Iteration

WANG FANGYU
MAT2309449

Introduction

In this report, we will learn the principles behind how the power iteration method is implemented to compute the eigenvector corresponding to positive dominant eigenvalue of a matrix.

In linear algebra, the power iteration, or power method, is also known as the Von Mises iteration, and it is an eigenvalue algorithm. It plays an important role in many real-world problems. For instants, in quantum mechanics, the power iteration can be used to find the ground state energy of a system by computing the largest eigenvalue of the Hamiltonian matrix. The power iteration is also the basis of Google's PageRank algorithm. PageRank determines the relative importance of web pages by computing the eigenvector of the transition matrix of the web graph. This is crucial in search engines and other graph-based ranking algorithms.

Dominant Eigenvalue: If the distinct eigenvalues of a matrix A are $\lambda_1, \lambda_2, ..., \lambda_k$, and if $|\lambda_1|$ is larger than $|\lambda_2|, ..., |\lambda_k|$, then λ_1 is called a dominant eigenvalue of A. Any eigenvector corresponding to a dominant eigenvalue is called a dominant. eigenvector of A.

The main purpose of this report is to investigate the another well-known relation, given as follows:

Theorem 1. *Let A be a symmetricn $\times$ n matrix that has a positive dominant eigenvalue λ. If x_0 is a unit vector in $\mathbb{R}^n$ that is not orthogonal to the eigenspace corresponding to λ, then the normalized power sequence*

$$x_0, x_1 = \frac{Ax_0}{\|Ax_0\|}, x_2 = \frac{Ax_1}{\|Ax_1\|}, ..., x_k = \frac{Ax_{k-1}}{\|Ax_{k-1}\|}, ...$$

converges to a unit dominant eigenvector.

We will investigate the convergence of the sequence in two manners. In the coming section, we present a numerical computation through Python to compute the first few terms of the sequence. In the last section, we then present a mathematical discussion on the convergence.

Numerical evidence of convergence

Let's consider an intuitive example where we set up a symmetric matrix A:

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

The unit dominant eigenvector of A(four decimal places):

$$v_1 \approx \begin{bmatrix} -0.1760 \\ 0.3063 \\ 0.6193 \\ -0.7012 \end{bmatrix}$$

Then we start the sequence as:

$$x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Apparently, x_0 is not orthogonal to v_1.

To visually demonstrate the convergence of sequence x_k towards the dominant eigenvector, we define the distance sequenceD_k:

$$D_k = \sqrt{\sum_{i=1}^{4} |x_{ki} - v_{1i}|^2}$$

The following is the first 10 terms of D_k sequence:

$$D_1 = \sqrt{\sum_{i=1}^{4} |x_{1i} - v_{1i}|^2} \approx 0.5319 D_2 = \sqrt{\sum_{i=1}^{4} |x_{2i} - v_{1i}|^2} \approx 0.1961 D_3 = \sqrt{\sum_{i=1}^{4} |x_{3i} - v_{1i}|^2} \approx 0.0509$$

$$D_4 = \sqrt{\sum_{i=1}^{4} |x_{4i} - v_{4i}|^2} \approx 0.0146 D_5 = \sqrt{\sum_{i=1}^{4} |x_{5i} - v_{1i}|^2} \approx 0.0042 D_6 = \sqrt{\sum_{i=1}^{4} |x_{6i} - v_{1i}|^2} \approx 0.0012$$

$$D_7 = \sqrt{\sum_{i=1}^{4} |x_{7i} - v_{1i}|^2} \approx 0.0003 D_8 = \sqrt{\sum_{i=1}^{4} |x_{8i} - v_{1i}|^2} \approx 0.0001 D_9 = \sqrt{\sum_{i=1}^{4} |x_{9i} - v_{1i}|^2} \approx 0.0000$$

$$D_{10} = \sqrt{\sum_{i=1}^{4} |x_{10i} - v_{1i}|^2} \approx 0.0000$$

As seen, actually from 6th term the D_k sequence is already very close to 0 and actually the 9th and 10th terms have reached magnitudes of 10^{-5} and 10^{-6}, respectively.

Now we provide a numerical scheme to test the convergence. In this Python program, it computes recursively the sequence D_k, and plot the terms (n versus D_k) in a graph.

```
import numpy as np
import matplotlib.pyplot as plt

A = np.array([[-1, 2, -3, 4],
              [2, -5, 6, 7],
              [-3, 6, 8, -9],
              [4, -7, -9, 10]])

eig_vals, eig_vecs = np.linalg.eig(A)

max_index = np.argmax(eig_vals)
l = eig_vecs[:, max_index]
l = l / np.linalg.norm(l)

x0 = np.array([1, 0, 0, 0])

iterations = 20

distance_to_l = np.zeros(iterations)

x = x0
for iter in range(iterations):
    x = A @ x
    x = x / np.linalg.norm(x)
```

```
    distance_to_l[iter] = np.linalg.norm(x - l)


plt.figure()
plt.plot(range(1, iterations + 1), distance_to_l, '-o')
plt.xlabel('Iterations')
plt.ylabel('The distance of the vector and the dominant eigenvector')
plt.title('The change of iteration number and vector distance')

plt.xticks(range(1, iterations + 1))
plt.grid(True)

plt.savefig('D:/power_iteration_distance.png')

plt.show()
```

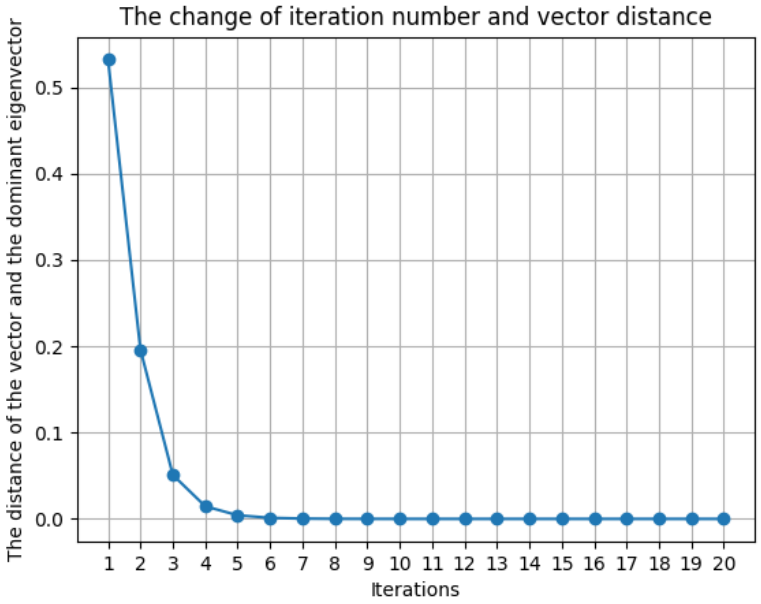

Figure 1: The first 20 terms from D_k.

Analytic computation

Let A be a symmetric $n \times n$ matrix which has the sorted in descending order eigenvalues and corresponding eigenvectors:

$$\lambda_1, \lambda_2, ..., \lambda_n$$
$$v_1, v_2, ..., v_n$$

Thus, λ_1 is the dominant eigenvalue of A,and v_1 is the corresponding eigenvector.
Then we start the power sequence as:

$$x_0 = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

which is a unit vector in $\mathbb{R}^n$ that is not orthogonal to the eigenspace corresponding to v_1, and could expressed as:

$$x_0 = c_1 v_1 + c_2 v_2 + ... + c_n v_n$$
$$(c_1 ... c_n \textit{ are scalar constants})$$

The normalized power sequence

$$x_0, x_1 = \frac{Ax_0}{\|Ax_0\|}, x_2 = \frac{Ax_1}{\|Ax_1\|}, ..., x_k = \frac{Ax_{k-1}}{\|Ax_{k-1}\|}, ...$$

can be represented in the following way:

$$x_k = \frac{Ax_{k-1}}{\|Ax_{k-1}\|} = \frac{A^k \cdot x_0}{\|A^k \cdot x_0\|}$$

Put the sight on $A^k \cdot x_0$, since $x_0 \in$ the eigenspace of A, we have:

$$A^k \cdot x_0 = (\lambda_1)^k c_1 v_1 + (\lambda_2)^k c_2 v_2 + ... + (\lambda_n)^k c_n v_n$$

$$\lim_{k \to \infty} A^k \cdot x_0 = \lim_{k \to \infty} (\lambda_1)^k c_1 v_1 + (\lambda_2)^k c_2 v_2 + ... + (\lambda_n)^k c_n v_n \tag{1}$$

Apparently, (1) is a vector and let us express it as v^*.

For vectors, multiplying by a scalar constant does not change its direction, so v^* has the same direction of $\frac{1}{{\lambda_1}^k} \cdot v^*$, which can be expressed as:

$$\frac{1}{{\lambda_1}^k} \cdot v^* = \lim_{k \to \infty} c_1 v_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k c_2 v_2 + ... + \left(\frac{\lambda_n}{\lambda_1}\right)^k c_n v_n \tag{2}$$

Since:

$$|\lambda_1| > |\lambda_2| \geqslant |\lambda_3| \geqslant ... \geqslant |\lambda_n|$$

Thus:

$$\lim_{k\to\infty} \left(\frac{\lambda_2}{\lambda_1}\right)^k = \lim_{k\to\infty} \left(\frac{\lambda_3}{\lambda_1}\right)^k = ... \lim_{k\to\infty} \left(\frac{\lambda_n}{\lambda_1}\right)^k = 0$$

$$\frac{1}{{\lambda_1}^k} \cdot v^* = \lim_{k\to\infty} c_1 v_1 = c_1 v_1$$

Thus v^* has the same direction of v_1.

$$\lim_{k\to\infty} x_k = v^* \cdot \frac{1}{\|A_k \cdot x_0\|}$$

Thus $\lim_{k\to\infty} x_k$ has the same direction of v_1.

We know that x_k and the eigenvector v_1 is both normalized, which mean the length of x_k and v_1 are both 1.

Thus we have:

$$\left\| \lim_{k\to\infty} x_k \right\| = \|v_1\|$$

$$\frac{\lim_{k\to\infty} x_k}{\|\lim_{k\to\infty} x_k\|} = \frac{v_1}{\|v_1\|}$$

Thus:

$$\lim_{k\to\infty} x_k = v_1$$