Chapter 5

The Power Method

The power method is a simple technique for finding individual eigenvalues and eigenvectors. It can be used on both symmetric and unsymmetric matrices.

Let $A \in \mathbb{C}^{n\times n}$ have eigenvalues $\lambda_1, \ldots, \lambda_n$ ordered so that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

The assumption that A has a distinct largest eigenvalue is essential. Assume that there is a similarity transformation of A so that

$$D = X^{-1}AX = \mathrm{diag}(\lambda_1,\, \lambda_2, \ldots,\, \lambda_n).$$

Such a matrix exists for most matrices (exercise). The columns of X are the eigenvectors of A. Let v be an arbitrary vector and consider the sequence

$$v,\ Av,\ A^2v,\ A^3v, \ldots.$$

As X is nonsingular its columns form a basis for $\mathbb{C}^n$ so v can be expressed as a linear combination of them. Writing $X = \left[\ x_1 \mid x_2 \mid \ldots \mid x_n\ \right]$, the vector v can be written as the linear combination

$$v = \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_n x_n,$$

hence

$$\begin{aligned} Av &= A(\gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_n x_n) \\ &= \gamma_1 Ax_1 + \gamma_2 Ax_2 + \cdots + \gamma_n Ax_n \\ &= \gamma_1\lambda_1 x_1 + \gamma_2\lambda_2 x_2 + \cdots + \gamma_n\lambda_n x_n. \end{aligned}$$

Similarly,

$$\begin{aligned} A^2v &= \gamma_1\lambda_1^2 x_1 + \cdots + \gamma_n\lambda_n^2 x_n, \\ A^3v &= \gamma_1\lambda_1^3 x_1 + \cdots + \gamma_n\lambda_n^3 x_n, \\ &\ \vdots \\ A^kv &= \gamma_1\lambda_1^k x_1 + \cdots + \gamma_n\lambda_n^k x_n. \end{aligned}$$

Now since λ_1 is the biggest of the eigenvalues, the term $\gamma_1\lambda_1^k x_1$ will begin to dominate all the other terms in the expression. If we write

$$A^kv = \lambda_1^k\left[\gamma_1 x_1 + \gamma_2\left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 + \cdots + \gamma_n\left(\frac{\lambda_n}{\lambda_1}\right)^k x_n\right],$$

then as k increases $(\lambda_i/\lambda_1)^k \to 0$ $(i > 1)$, and we see that $A^k v$ points increasingly in the direction of the dominant eigenvector of A. To see this more clearly, let $q_k = \frac{1}{\lambda_1^k} A^k v$. Then,

$$\begin{aligned}
\|q_k - \gamma_1 x_1\| &= \left\| \gamma_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 + \cdots + \gamma_n \left(\frac{\lambda_n}{\lambda_1}\right)^k x_n \right\| \\
&\leq |\gamma_2| \left|\frac{\lambda_2}{\lambda_1}\right|^k \|x_2\| + \cdots + |\gamma_n| \left|\frac{\lambda_n}{\lambda_1}\right|^k \|x_n\| \\
&\leq (|\gamma_2|\|x_2\| + \cdots + |\gamma_n|\|x_n\|) \left|\frac{\lambda_2}{\lambda_1}\right|^k \\
&= C \left|\frac{\lambda_2}{\lambda_1}\right|^k \to 0 \quad \text{as } k \to \infty.
\end{aligned}$$

The rate of convergence of q_k to $\gamma_1 x_1$ is linear: as $k \to \infty$ the ratio of the errors

$$\frac{\|q_{k+1} - \gamma_1 x_1\|}{\|q_k - \gamma_1 x_1\|}$$

tends to the constant limit $|\lambda_2/\lambda_1|$.

5.1 A Practical Algorithm

Since $A^k v$ tends to point in the direction of an eigenvector of A, the following algorithm could be used.

Algorithm 5.1: Power method 1
Choose an initial vector v_0
for $k = 1, 2, \ldots$
    $v_k = A v_{k-1}$
end

The problem is that the vector v_k can become unboundedly large (if $|\lambda_1| > 1$) or unboundedly small (if $|\lambda_1| < 1$). In practice, as in our analysis, we introduce a scaling. This gives us the following algorithm.

Algorithm 5.2: Power method 2
Choose an initial vector v_0
for $k = 1, 2, \ldots$
    $z_k = A v_{k-1}$
    $v_k = z_k / \sigma_k$
end

Here $\sigma_1, \sigma_2, \ldots$ is a sequence of scaling factors. The precise choice of scaling factors is not particularly important—if x is an eigenvector then so is αx for any scalar $\alpha \neq 0$—it is only the direction of the vector that really matters. A convenient choice of scaling factors is $\sigma_k = \|z_k\|_2$. This ensures that $\|v_k\|_2 = 1$ for all values of k, and the problem of unboundedness will not arise. Notice that as the sequence converges the vector v_k begins to resemble the dominant eigenvector x_1 hence

$$z_{k+1} = A v_k \to \lambda_1 v_k.$$

Also

$$r_{k+1} = v_k^T A v_k = v_k^T z_{k+1} \to \lambda_1 v_k^T v_k = \lambda_1.$$

The quantity r_k is known as the *Rayleigh quotient* of v_k and gives an estimate to the eigenvalue on each step.

The practical power method takes all this into account. It can be written as follows.

Algorithm 5.3: The practical power method
Choose TOL and v_0, let $r_0 = 0$, $k = 0$
repeat
  $k = k + 1$
  $z_k = Av_{k-1}$
  $r_k = v_{k-1}^T z_k$
  $v_k = z_k / \|z_k\|_2$
until $|r_k - r_{k-1}| < TOL$
$\lambda = r_k$, $x = v_k$

Example 5.1.1 *Let*

$$A = \begin{bmatrix} 10 & 2 \\ 1 & 1 \end{bmatrix}, \quad v_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Then

$$z_1 = Av_0 = \begin{bmatrix} 10 \\ 1 \end{bmatrix}, \quad r_1 = 10, \quad v_1 = \begin{bmatrix} 0.9950 \\ 0.0995 \end{bmatrix},$$

and

$$z_2 = \begin{bmatrix} 10.1494 \\ 1.0945 \end{bmatrix}, \quad r_2 = 10.2079, \quad v_2 = \begin{bmatrix} 0.9942 \\ 0.1072 \end{bmatrix}.$$

The following table shows how the iteration progresses

k	r_k	$v_k(1)$	$v_k(2)$
3	10.2163	0.9942	0.1078
4	10.2169	0.9942	0.1079
5	10.2170	0.9942	0.1079
6	10.2170	0.9942	0.1079

After six iterations the iteration has converged and we conclude that the largest eigenvalue of A is 10.2170 with eigenvector $[0.9942 \quad 0.1079]^T$.

5.1.1 Efficient Implementation

If we look at the rate of convergence of v_k to x_1 we find the ratio

$$\frac{\|v_{k+1} - x_1\|}{\|v_k - x_1\|}$$

tends towards a limit, namely $|\lambda_2/\lambda_1|$. So for the power method to be effective it is essential that its biggest eigenvalue is significantly bigger than any of the others. In practice, this is not often true, and we must use more effective methods.

Each step of the power method is cheap to perform, requiring only one matrix-vector multiply ($O(n^2)$ operations) and a normalisation ($O(n)$ operations).

In our analysis we wrote our initial vector v as

$$v = \gamma_1 x_1 + \cdots \gamma_n x_n.$$

If $\gamma_1 = 0$ then the iteration will not converge to (λ_1, cx_1). This is not a problem in practice, however, since

1. The probability of choosing v at random so that $\gamma_1 = 0$ is negligible.
2. If, by some fluke, γ_1 was equal to zero rounding errors would introduce a contribution from x_1 into our iterates, and the method will work as normal.

5.2 Extending The Power Method

The power method in its simplest form can only be used to compute the dominant eigenpair of a matrix. Using some simple properties of a matrix's eigensystem we will now develop algorithms to compute the other eigenvalues and eigenvectors.

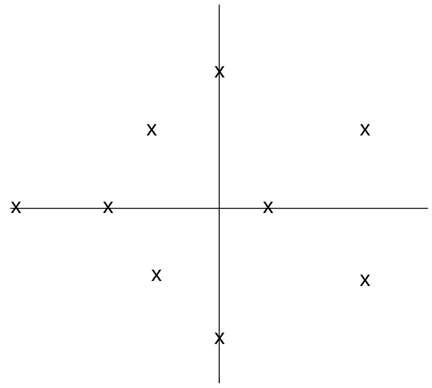
5.2.1 Shifts

Suppose A has eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding eigenvectors $x_1, \ldots, x_n$. Then the matrix $A - \rho I$ has eigenvalues $\lambda_1 - \rho, \ldots, \lambda_n - \rho$ and eigenvectors $x_1, \ldots, x_n$. We can use this shifting property to derive a power method that converges to an alternative eigenvalue.

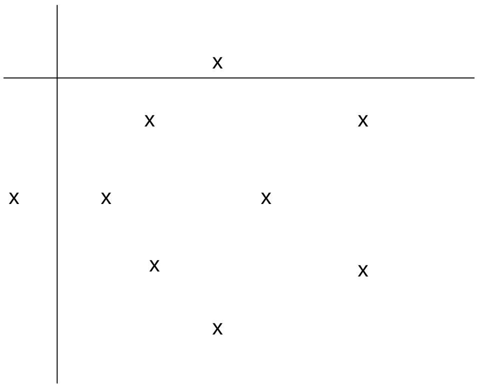
For example, suppose that A has eigenvalues 32, 16, 5, 0.3, -1 and -5. The regular power method will find the eigenvalue 32. If we use the power method on $A - 20I$, though, we converge to the largest eigenvalue of this matrix, namely -25. Using this information we find the eigenvalue -5 of A and its corresponding eigenvector. The shifted power method can be written as follows.

Algorithm 5.4: The shifted power method
Choose TOL, v_0 and $\rho \in \mathbb{C}$, let $r_0 = 0$, $k = 0$
repeat
    $k = k + 1$
    $z_k = (A - \rho I)v_{k-1}$
    $r_k = v_{k-1}^T z_k$
    $v_k = z_k / \|z_k\|_2$
until $|r_k - r_{k-1}| < TOL$
$\lambda = r_k + \rho$, $x = v_k$

The problem with the shifted power method is that it can only be used to find a select few of A's eigenvalues. The eigenvalues lie scattered around the complex plane:



The effect of a shift is to "shift" the axes:



Only a select few of the eigenvalues can ever be made to be the largest by this transformation (Exercise: which ones?), so the shifted power method can still only be used to find a few eigenvalues of A.

5.2.2 Inverse Iteration

If A has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ then A^{-1} has eigenvalues $1/\lambda_1, 1/\lambda_2, \ldots, 1/\lambda_n$ (and the same eigenvectors as A). Therefore if we use the power method on A^{-1} we will converge to $1/\lambda_n$, from which we can find A's smallest eigenvalue. Whilst shifting can only be used to make a handful of eigenvalues big (relative to the others), by making an appropriate shift ρ we can make $\lambda_j - \rho$ the smallest eigenvalue of $A - \rho I$ *for all* λ_j (choose $\rho \approx \lambda_j$). So, by using the power method on $(A - \rho I)^{-1}$ we can in theory find all the eigenvalues of A.

In practice we do not form $(A - \rho I)^{-1}$. Instead, to find z_k we solve $(A - \rho I) z_k = v_{k-1}$. If we find the LU decomposition of $A - \rho I$ during the first iteration (at a cost of $O(n^3)$ flops) we can reuse it on all future iterations (which means each successive step costs only $O(n^2)$ flops). The algorithm for inverse iteration can be written as follows.

Algorithm 5.5: Inverse iteration
Choose TOL, v_0 and $\rho \in \mathbb{C}$, let $r_0 = 0$, $k = 0$
Solve $LU = A - \rho I$
repeat
  $k = k + 1$
  Solve $(A - \rho I) z_k = v_{k-1}$
  $r_k = v_{k-1}^T z_k$
  $v_k = z_k / \|z_k\|_2$
until $|r_k - r_{k-1}| < TOL$
$\lambda = 1/r_k + \rho$, $x = v_k$

5.2.3 Practical Details

If $\rho \approx \lambda_j$ then $1/(\lambda_j - \rho)$ is likely to be a lot larger than $1/(\lambda_i - \rho)$ for all $i \neq j$, hence for good choices of ρ inverse iteration will converge rapidly. To pick a good value of ρ, though, requires some prior knowledge of the eigenvalues. For this reason inverse iteration is generally used when an estimate of an eigenvalue has been obtained from another algorithm. In these situations inverse iteration can be used to improve the eigenvalue estimate as well as compute the associated eigenvector.

If A is tridiagonal then we can solve $(A - \rho I) z_{k+1} = v_k$ in just $O(n)$ operations. Using the algorithms of the previous chapter we can obtain excellent eigenvalue estimates. We then only need a couple of steps of inverse iteration to find each eigenvector. This is what is normally done in practice.

Another point to note is that if ρ is close to an eigenvalue of A then $A - \rho I$ is close to being singular so it will be ill-conditioned. Now ill-conditioned linear systems are highly sensitive to the errors caused by computer arithmetic so it might be thought that the vectors $\{z_k\}$ will be computed inaccurately. Conveniently, though, the errors are concentrated in the direction of the eigenvector, and they cause little problem.

Example 5.2.1 *We return to Example 5.1.1 using inverse iteration with* $\rho = 10$ *(if a matrix has a large diagonal element then this will be a good approximation to an actual eigenvalue). Hence,*

$$A - 10I = \begin{bmatrix} 0 & 2 \\ 1 & -9 \end{bmatrix}, \quad v_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Then

$$z_1 = (A - 10I)^{-1} v_0 = \begin{bmatrix} 4.5 \\ 0.5 \end{bmatrix}, \quad r_1 = 4.5, \quad v_1 = \begin{bmatrix} 0.9939 \\ 0.1104 \end{bmatrix},$$

and the iteration progresses as follows.

k	r_k	$v_k(1)$	$v_k(2)$
2	4.6098	0.9942	0.1078
3	4.6085	0.9942	0.1079
4	4.6085	0.9942	0.1079

And $\lambda = \rho + 1/r_4 = 10.2170$.