Recall: The standard way of computing eigenvalues and eigenvectors of an $n \times n$ matrix A:

1) Compute the characteristic polynomial $P(\lambda) = \det(A - \lambda I_n)$.

2) Eigenvalues of A = roots of $P(\lambda)$.

3) Eigenvectors corresponding to an eigenvalue λ = vectors in $\mathrm{Nul}(A - \lambda I_n)$.

Problems:

- For large matrices computations of $P(\lambda)$ are slow.
- Even if we know $P(\lambda)$, it is difficult to compute its roots.

More efficient way: The power method.

Assumptions:

- A is an $n \times n$ matrix with eigenvalues $\lambda_1, \dots, \lambda_n$ such that

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$$

- For $i = 1, \dots, n$, by $\mathbf{w}_i$ we will denote an eigenvector corresponding to the eigenvalue λ_i, such that $||\mathbf{w}_i|| = 1$.

Computing the largest eigenvalue λ_1 its eigenvector

- Start with a vector $v \neq 0$. Since A is diagonalizable we have $v = \sum_i c_i w_i$ for some $c_i \in \mathbb{R}$. If v is selected at random, then almost always we will have $c_1 \neq 0$.

- Denote: $v_k = \dfrac{A^k v}{\|A^k v\|}$

We have: $A^k v = A^k(\sum c_i w_i) = \sum c_i (A^k w_i) = \sum c_i \lambda_i^k w_i$

This gives:

$$v_k = \frac{\sum c_i \lambda_i^k w_i}{\|\sum c_i \lambda_i^k w_i\|} = \frac{\lambda_1^k \left(c_1 w_1 + \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k w_i\right)}{|\lambda_1|^k \cdot \left\|c_1 w_1 + \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k w_i\right\|}$$

Since $\left|\frac{\lambda_i}{\lambda_1}\right| < 1$ for $i \geq 2$ thus $\lim_{k\to\infty} \left(\frac{\lambda_i}{\lambda_1}\right)^k = 0$

Thus for large k we get

$$v_k \approx \frac{\lambda_1^k c_1 w_1}{|\lambda_1|^k \|c_1 w_1\|} = \frac{\lambda_1^k}{|\lambda_1|^k} \cdot \frac{c_1}{|c_1|} \cdot \frac{w_1}{\|w_1\|} \overset{\|w_1\| = 1}{=} \pm 1 \cdot w_1$$

Note: Both w_1 and $-w_1$ are eigenvectors corresponding to the eigenvalue λ_1, so it does not matter if we get $v_k \approx w_1$ or $v_k \approx -w_1$

Computations of λ_1

Since $v_k \approx \pm w_1$ thus $Av_k \approx \lambda_1 v_k$.
Dividing entries of Av_k by entries of v_k we get an approximated value of λ_1.

Note: In practice the vectors v_k are computed iteratively:

$$v_0 = v, \quad v_1 = \frac{Av_0}{\|Av_0\|}, \quad v_2 = \frac{Av_1}{\|Av_1\|}, \quad \dots, \quad v_k = \frac{Av_{k-1}}{\|Av_{k-1}\|}$$

Computing the other eigenvalues and eigenvectors

Idea: If we could start the power metod with a vector $v = \sum_i c_i w_i$ such that $c_1 = 0$ then the power method would compute w_2 and λ_2.

Problem: How to find such a vector v ?

Solution for a symmetric matrix A:

Recall: For a symmetric matrix eigenvectors for different eigenvalues are orthogonal:

$$w_i \cdot w_j = 0 \quad \text{for } i \neq j$$

- start with an arbitrary vector $v = \sum c_i w_i$
- use v and the power method to compute w_1.
- we have:

$$v \cdot w_1 = \left(\sum c_i w_i\right) \cdot w_1 = \sum c_i (w_i \cdot w_1)$$
$$= c_1 w_1 \cdot w_1 \underset{\|w_1\| = 1}{=} c_1$$

- take $v' = v - (v \cdot w_1) w_1 = \sum c_i w_i - c_1 w_1 = \sum_{i=2}^{n} c_i w_i$

  We can use the power metod starting with v' to compute w_2 and λ_2

Note: Due to rounding errors in the power method we need to make the vector v'_k orthogonal to w_1 at each iteration step.

We can continue this process iteratively to compute all other eigenvectors and their corresponding eigenvalues.