

5.8 Iterative estimater for egenverdier (og egenvektorer)

Anta at A er diagonaliserbar og $n \times n$. Vi har da en egenvektorbasis

$$\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n\}$$

for \mathbb{R}^n med egenverdier $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ slik at

$$A \vec{v}_j = \lambda_j \vec{v}_j$$

for $j = 1, 2, \dots, n$. Vi kan representere hver $\vec{x}_0 \in \mathbb{R}^n$ unikt ved

$$\vec{x}_0 = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \dots + c_n \vec{v}_n.$$

Altså blir

$$\vec{x}_1 = A \vec{x}_0 = c_1 \lambda_1 \vec{v}_1 + c_2 \lambda_2 \vec{v}_2 + \dots + c_n \lambda_n \vec{v}_n$$

\vdots

$$\vec{x}_k = A \vec{x}_{k-1} = c_1 \lambda_1^k \vec{v}_1 + c_2 \lambda_2^k \vec{v}_2 + \dots + c_n \lambda_n^k \vec{v}_n.$$

Anta videre at A har en **dominant** egenverdi, det vil si at $|\lambda_1| > |\lambda_j|$ når $\lambda_j \neq \lambda_1$. Da vil $|\lambda_1|^k$ vokse mye raskere enn $|\lambda_j|^k$, og

$$\vec{x}_k \approx c_1 \lambda_1^k \vec{v}_1.$$

Eksempel 5.8.1

La $A = \begin{bmatrix} 1.8 & 0.8 \\ 0.2 & 1.2 \end{bmatrix}$ og $\vec{x}_0 = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix}$. Regn ut $\vec{x}_1, \dots, \vec{x}_9$.

$$\vec{x}_1 = A \vec{x}_0 = \begin{bmatrix} 1.8 & 0.8 \\ 0.2 & 1.2 \end{bmatrix} \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.9 + 0.8 \\ -0.1 + 1.2 \end{bmatrix} = \begin{bmatrix} -0.1 \\ 1.1 \end{bmatrix}.$$

$$\vec{x}_2 = A \vec{x}_1 = \dots = \begin{bmatrix} 0.7 \\ 1.3 \end{bmatrix}$$

\vdots

$$\vec{x}_7 = A \vec{x}_6 = \dots = \begin{bmatrix} 50.3 \\ 17.7 \end{bmatrix}$$

$$\vec{x}_9 = A \vec{x}_8 = \dots = \begin{bmatrix} 101.5 \\ 26.5 \end{bmatrix}$$

Vi ser at $2 \begin{bmatrix} 50.3 \\ 17.7 \end{bmatrix} \approx \begin{bmatrix} 101.5 \\ 26.5 \end{bmatrix}$ og $101.5 \approx 4 \cdot 26.5$. Vi gjetter at $\lambda = 2$ er en egenverdi og $\vec{v} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$ en egenvektor.

Det er **plausibelt** for oss (og datamaskinen) at tallet i \vec{x}_k blir større og større. Dersom vi **visste** hva λ_1 er, kunne vi heller definert

$$\vec{x}_{k+1} = \frac{1}{\lambda_1} A \vec{x}_k = c_1 \vec{v}_1 + c_2 \underbrace{\left(\frac{\lambda_2}{\lambda_1}\right)^k}_{\rightarrow 0} \vec{v}_2 + \dots + c_n \underbrace{\left(\frac{\lambda_n}{\lambda_1}\right)^k}_{\rightarrow 0} \vec{v}_n$$

for å unngå problemet.

Generelt vil vi normalisere på en annen måte.

Algoritme (Powersmetoden)

- ① Velg \vec{x}_0 slik at den største absoluttverdien av tallet i \vec{x}_0 er lik 1.
- ② For $k=0, 1, \dots$
 - (a) Regn ut $A \vec{x}_k$.
 - (b) La μ_k være det tilkt i $A \vec{x}_k$ med størst absoluttverdi.
 - (c) Regn ut $\vec{x}_{k+1} = \frac{1}{\mu_k} A \vec{x}_k$.
- ③ Da vil μ_k være et estimat for λ_1 og \vec{x}_k være et estimat for \vec{v}_1 .

Hva kan gå galt?

- Startvektoren \vec{x}_0 ikke noe komponent i egenvektoren til λ_1 .
- To forskjellige egenverdier har samme og største absoluttverdi. F.eks. $\lambda_1=9$, $\lambda_2=2$ og $\lambda_3=-9$.