Approximation des valeurs et vecteurs propres d'une matrice, applications

S. Mottelet

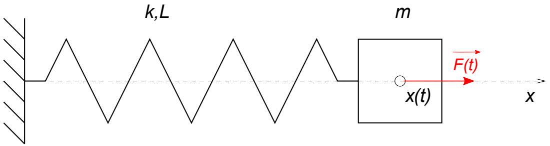
Université de Technologie de Compiègne

Plan

1. Motivations (1) : résonance mécanique
2. Méthodes numériques
   1. Algorithme de la puissance
   2. Algorithme de la puissance inverse
3. Motivations (2) : algorithme PageRank

Motivation

Résonance mécanique, Oscillations forcées d'un système masse-ressort

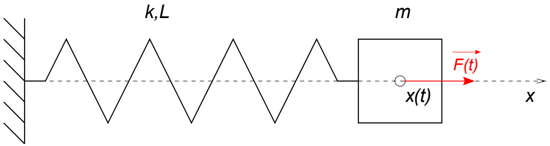


$$mx'' + k(x - L) = f(t),\ t > 0,$$
$$x(0) = L,$$
$$x'(0) = 0,$$

$$f(t) =< \overrightarrow{F(t)}, \vec{\imath} > .$$

Motivation

Résonance mécanique, Oscillations forcées d'un système masse-ressort



$$\begin{aligned} u'' + \omega_0^2 u &= \frac{1}{m} f(t),\ t > 0, \\ u(0) &= 0, \\ u'(0) &= 0, \end{aligned}$$

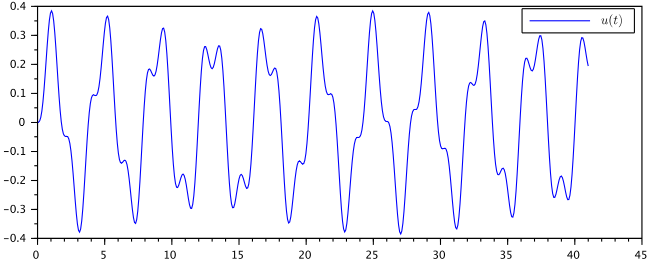
après avoir posé $u = (x - L)$ et $\omega_0 = \sqrt{\frac{k}{m}}$.

Motivation

Résonance mécanique, Oscillations forcées d'un système masse-ressort

Solutions particulières pour $f(t) = a \sin \omega t$:

- Si $\omega \neq \omega_0$, on a $u_p(t) = \dfrac{a}{m} \dfrac{\sin \omega t}{\omega_0^2 - \omega^2}$, $u(t) = \frac{a}{m(\omega_0^2 - \omega^2)} \left(\sin \omega t - \frac{\omega}{\omega_0} \sin \omega_0 t \right)$.



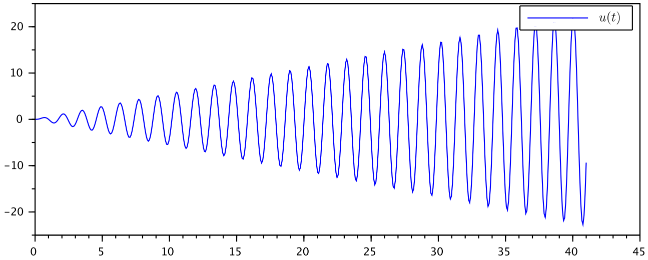
la solution $t \to u(t)$ est bornée pour $t > 0$.

Motivation

Résonance mécanique, Oscillations forcées d'un système masse-ressort

Solutions particulières pour $f(t) = a \sin \omega t$:

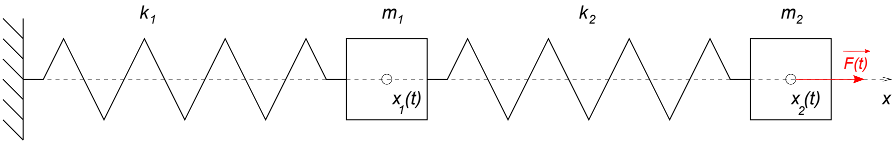
- Si $\omega = \omega_0$, on a $u_p(t) = -\dfrac{a}{m}\dfrac{t\cos\omega_0 t}{2\omega_0}$, $u(t) = \frac{a}{2\omega_0 m}\left(-t\cos\omega_0 t + \frac{1}{\omega_0}\sin\omega_0 t\right)$.



et $t \to u(t)$ est non-bornée pour $t > 0$, le système entre en résonance.

Motivation

Résonance mécanique, système à deux degrés de liberté



On pose $u_1 = x_1 - L_1$, $u_2 = x_2 - L_1 - L_2$ (L_1, L_2 : longueurs au repos des ressorts)

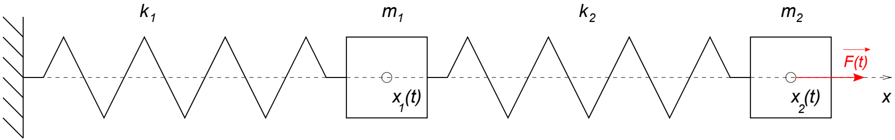
$$\left\{\begin{array}{rcl} m_1 u_1'' + (k_1 + k_2)u_1 - k_2 u_2 & = & 0, \\ m_2 u_2'' - k_2 u_1 + k_2 u_2 & = & f(t), \end{array}\right.$$

$$\mathbf{u}'' + \mathbf{A}\mathbf{u} = \mathbf{b} f(t), \quad \mathbf{A} = \begin{pmatrix} \frac{k_1+k_2}{m_1} & -\frac{k_2}{m_1} \\ -\frac{k_2}{m_2} & \frac{k_2}{m_2} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 1/m_2 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

Hypothèse : $\mathbf{A}$ diagonalisable, $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$, $\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$.

Motivation

Résonance mécanique, système à deux degrés de liberté



On pose $u_1 = x_1 - L_1$, $u_2 = x_2 - L_1 - L_2$ (L_1, L_2 : longueurs au repos des ressorts)

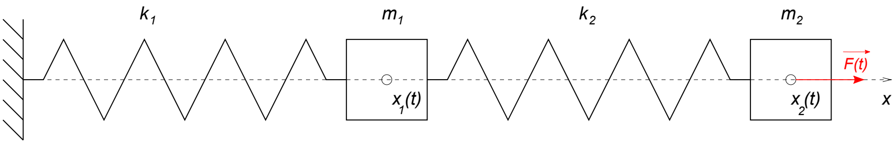
$$\left\{\begin{array}{rcl} m_1 u_1'' + (k_1 + k_2)u_1 - k_2 u_2 & = & 0, \\ m_2 u_2'' - k_2 u_1 + k_2 u_2 & = & f(t), \end{array}\right.$$

$$\mathbf{u}'' + \mathbf{A}\mathbf{u} = \mathbf{b} f(t), \quad \mathbf{A} = \begin{pmatrix} \frac{k_1+k_2}{m_1} & -\frac{k_2}{m_1} \\ -\frac{k_2}{m_2} & \frac{k_2}{m_2} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 1/m_2 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

Hypothèse : $\mathbf{A}$ diagonalisable, $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$, $\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$.

Motivation

Résonance mécanique, système à deux degrés de liberté



On pose $u_1 = x_1 - L_1$, $u_2 = x_2 - L_1 - L_2$ (L_1, L_2 : longueurs au repos des ressorts)

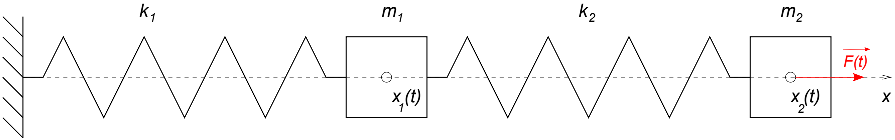
$$\begin{cases} m_1 u_1'' + (k_1 + k_2)u_1 - k_2 u_2 &= 0, \\ m_2 u_2'' - k_2 u_1 + k_2 u_2 &= f(t), \end{cases}$$

$$\mathbf{u}'' + \mathbf{A}\mathbf{u} = \mathbf{b}f(t), \quad \mathbf{A} = \begin{pmatrix} \frac{k_1+k_2}{m_1} & -\frac{k_2}{m_1} \\ -\frac{k_2}{m_2} & \frac{k_2}{m_2} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 1/m_2 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

Hypothèse : $\mathbf{A}$ diagonalisable, $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$, $\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$.

Motivation

Résonance mécanique, système à deux degrés de liberté



On pose $\mathbf{u} = \mathbf{P}\mathbf{v}$, $\mathbf{b} = \mathbf{P}\mathbf{c}$:

$$\mathbf{u}'' + \mathbf{A}\mathbf{u} = \mathbf{b}f(t) \Leftrightarrow \mathbf{P}\mathbf{v}'' + \mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{P}\mathbf{c}f(t),$$

$$\Leftrightarrow \mathbf{v}'' + \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{c}f(t),$$

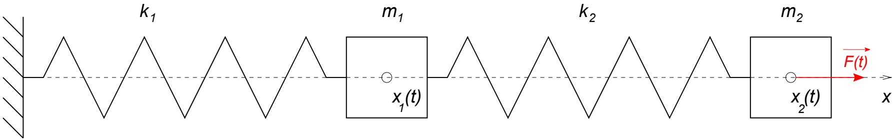
$$\Leftrightarrow \mathbf{v}'' + \mathbf{D}\mathbf{v} = \mathbf{c}f(t),$$

$$\Leftrightarrow \left\{ \begin{array}{rcl} v_1'' + \lambda_1 v_1 & = & c_1 f(t), \\ v_2' + \lambda_2 v_2 & = & c_2 f(t), \end{array} \right.$$

Deux pulsations propres : $\omega_1 = \sqrt{\lambda_1}$, $\omega_2 = \sqrt{\lambda_2}$.

Motivation

Résonance mécanique, système à deux degrés de liberté



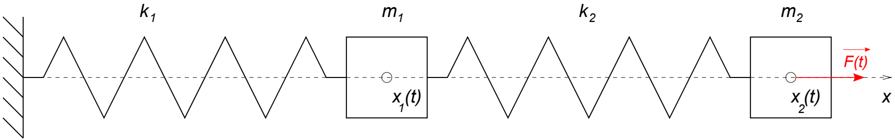
On pose $\mathbf{u} = \mathbf{P}\mathbf{v}$, $\mathbf{b} = \mathbf{P}\mathbf{c}$:

$$\begin{aligned}
\mathbf{u}'' + \mathbf{A}\mathbf{u} = \mathbf{b}f(t) &\Leftrightarrow \mathbf{P}\mathbf{v}'' + \mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{P}\mathbf{c}f(t),\\
&\Leftrightarrow \mathbf{v}'' + \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{c}f(t),\\
&\Leftrightarrow \mathbf{v}'' + \mathbf{D}\mathbf{v} = \mathbf{c}f(t),\\
&\Leftrightarrow \begin{cases} v_1'' + \lambda_1 v_1 &= c_1 f(t),\\ v_2' + \lambda_2 v_2 &= c_2 f(t), \end{cases}
\end{aligned}$$

Deux pulsations propres : $\omega_1 = \sqrt{\lambda_1}$, $\omega_2 = \sqrt{\lambda_2}$.

Motivation

Résonance mécanique, système à deux degrés de liberté



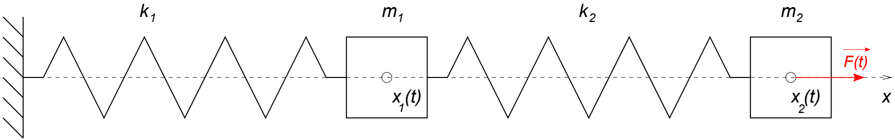
On pose $\mathbf{u} = \mathbf{Pv}$, $\mathbf{b} = \mathbf{Pc}$:

$$\begin{aligned}
\mathbf{u}'' + \mathbf{Au} = \mathbf{b}f(t) &\Leftrightarrow \mathbf{Pv}'' + \mathbf{APv} = \mathbf{Pc}f(t),\\
&\Leftrightarrow \mathbf{v}'' + \mathbf{P}^{-1}\mathbf{APv} = \mathbf{c}f(t),\\
&\Leftrightarrow \mathbf{v}'' + \mathbf{Dv} = \mathbf{c}f(t),\\
&\Leftrightarrow \left\{\begin{array}{lll} v_1'' + \lambda_1 v_1 & = & c_1 f(t),\\ v_2' + \lambda_2 v_2 & = & c_2 f(t),\end{array}\right.
\end{aligned}$$

Deux pulsations propres : $\omega_1 = \sqrt{\lambda_1}$, $\omega_2 = \sqrt{\lambda_2}$.

Motivation

Résonance mécanique, système à deux degrés de liberté



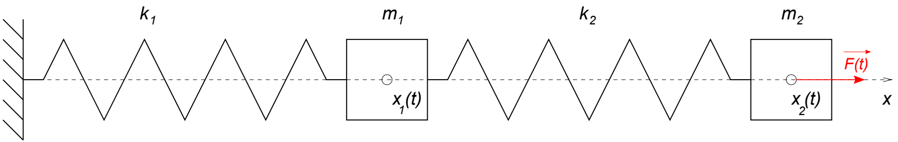
On pose $\mathbf{u} = \mathbf{P}\mathbf{v}$, $\mathbf{b} = \mathbf{P}\mathbf{c}$:

$$
\begin{aligned}
\mathbf{u}'' + \mathbf{A}\mathbf{u} = \mathbf{b}f(t) &\Leftrightarrow \mathbf{P}\mathbf{v}'' + \mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{P}\mathbf{c}f(t), \\
&\Leftrightarrow \mathbf{v}'' + \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{c}f(t), \\
&\Leftrightarrow \mathbf{v}'' + \mathbf{D}\mathbf{v} = \mathbf{c}f(t), \\
&\Leftrightarrow \left\{ \begin{array}{rcl} v_1'' + \lambda_1 v_1 & = & c_1 f(t), \\ v_2' + \lambda_2 v_2 & = & c_2 f(t), \end{array} \right.
\end{aligned}
$$

Deux pulsations propres : $\omega_1 = \sqrt{\lambda_1}$, $\omega_2 = \sqrt{\lambda_2}$.

Motivation

Résonance mécanique, système à deux degrés de liberté



On pose $\mathbf{u} = \mathbf{P}\mathbf{v}$, $\mathbf{b} = \mathbf{P}\mathbf{c}$:

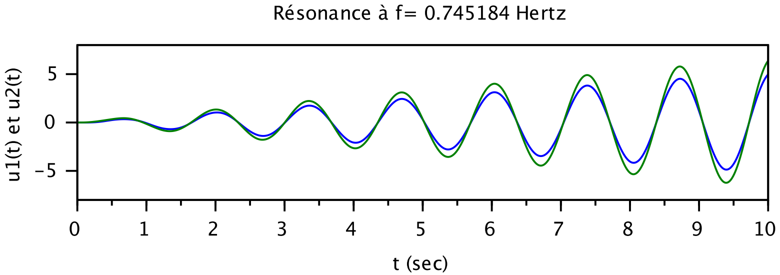
$$\begin{aligned}
\mathbf{u}'' + \mathbf{A}\mathbf{u} = \mathbf{b}f(t) &\Leftrightarrow \mathbf{P}\mathbf{v}'' + \mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{P}\mathbf{c}f(t),\\
&\Leftrightarrow \mathbf{v}'' + \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{v} = \mathbf{c}f(t),\\
&\Leftrightarrow \mathbf{v}'' + \mathbf{D}\mathbf{v} = \mathbf{c}f(t),\\
&\Leftrightarrow \left\{\begin{array}{rcl} v_1'' + \lambda_1 v_1 &=& c_1 f(t),\\ v_2' + \lambda_2 v_2 &=& c_2 f(t), \end{array}\right.
\end{aligned}$$

Deux pulsations propres : $\omega_1 = \sqrt{\lambda_1}$, $\omega_2 = \sqrt{\lambda_2}$.

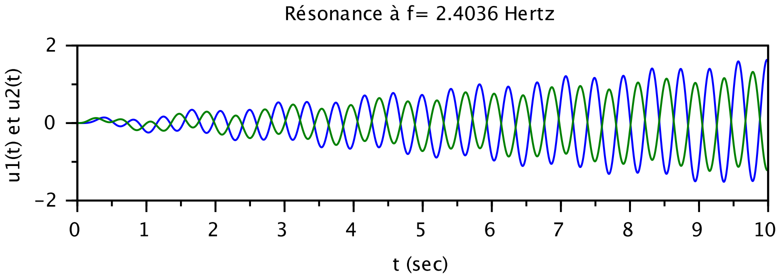
Motivation

Résonance mécanique, système à deux degrés de liberté

$$f(t) = a \sin \omega_1 t$$



$$f(t) = a \sin \omega_2 t$$



Motivation

Résonance mécanique, système à deux degrés de liberté

Démo Scilab

Soit $\mathbf{A}$ une matrice réelle $n \times n$, diagonalisable et de valeurs propres $(\lambda_i)_{i=1\dots n}$ avec $|\lambda_1| > |\lambda_2| \geq \dots |\lambda_n|$ et $\mathbf{y}_1, \dots, \mathbf{y}_n$ les vecteurs propres associés, vérifiant $\|\mathbf{y}_i\| = 1$.

Remarque : λ_1 est nécessairement simple et réelle

Théorème

Soit $\mathbf{x}_0 \notin \mathrm{Vect}(\mathbf{y}_2, \dots, \mathbf{y}_n)$. La suite $(\mathbf{x}_k)$ définie par

$$\mathbf{x}_{k+1} = \frac{\mathbf{A}\mathbf{x}_k}{\|\mathbf{A}\mathbf{x}_k\|},$$

converge dans le sous-espace propre associé à $\mathbf{y}_1$ au sens où

$$\lim_{k \to \infty} \mathrm{signe}(\lambda_1)^k \mathbf{x}_k = \pm \mathbf{y}_1.$$

Soit $\mathbf{A}$ une matrice réelle $n \times n$, diagonalisable et de valeurs propres $(\lambda_i)_{i=1\ldots n}$ avec $|\lambda_1| > |\lambda_2| \geq \ldots |\lambda_n|$ et $\mathbf{y}_1, \ldots, \mathbf{y}_n$ les vecteurs propres associés, vérifiant $\|\mathbf{y}_i\| = 1$.

Remarque : λ_1 est nécessairement simple et réelle

Théorème

Soit $\mathbf{x}_0 \notin \mathrm{Vect}(\mathbf{y}_2, \ldots, \mathbf{y}_n)$. La suite $(\mathbf{x}_k)$ définie par

$$\mathbf{x}_{k+1} = \frac{\mathbf{A}\mathbf{x}_k}{\|\mathbf{A}\mathbf{x}_k\|},$$

converge dans le sous-espace propre associé à $\mathbf{y}_1$ au sens où

$$\lim_{k\to\infty} \mathrm{signe}(\lambda_1)^k \mathbf{x}_k = \pm\mathbf{y}_1.$$

Démonstration : on peut tout d'abord montrer par récurrence que $\mathbf{x}_k = \dfrac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|}$, puis en écrivant $\mathbf{x}_0 = \sum_{i=1}^n \alpha_i \mathbf{y}_i$, (note : on a $\alpha_1 \neq 0$)

$$\mathbf{A}^k\mathbf{x}_0 = \sum_{i=1}^n \alpha_i \mathbf{A}^k \mathbf{y}_i = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{y}_i,$$

$$= \alpha_1 \lambda_1^k \mathbf{y}_1 + \sum_{i=2}^n \alpha_i \lambda_i^k \mathbf{y}_i,$$

$$= \alpha_1 \lambda_1^k \left(\mathbf{y}_1 + \sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k \mathbf{y}_i \right),$$

d'où

$$\lim_{k\to\infty} \left(\frac{|\lambda_1|}{\lambda_1} \right)^k \frac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|} = \frac{\alpha_1}{|\alpha_1|} \mathbf{y}_1.$$

Méthodes numériques

Méthode de la puissance

Démonstration : on peut tout d'abord montrer par récurrence que $\mathbf{x}_k = \dfrac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|}$, puis en écrivant $\mathbf{x}_0 = \sum_{i=1}^n \alpha_i \mathbf{y}_i$, (note : on a $\alpha_1 \neq 0$)

$$
\begin{aligned}
\mathbf{A}^k\mathbf{x}_0 &= \sum_{i=1}^n \alpha_i \mathbf{A}^k \mathbf{y}_i = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{y}_i,\\
&= \alpha_1 \lambda_1^k \mathbf{y}_1 + \sum_{i=2}^n \alpha_i \lambda_i^k \mathbf{y}_i,\\
&= \alpha_1 \lambda_1^k \left(\mathbf{y}_1 + \sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{y}_i \right),
\end{aligned}
$$

d'où

$$
\lim_{k\to\infty} \left(\frac{|\lambda_1|}{\lambda_1}\right)^k \frac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|} = \frac{\alpha_1}{|\alpha_1|}\mathbf{y}_1.
$$

Méthodes numériques

Méthode de la puissance

Démonstration : on peut tout d'abord montrer par récurrence que $\mathbf{x}_k = \dfrac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|}$, puis en écrivant $\mathbf{x}_0 = \sum_{i=1}^n \alpha_i \mathbf{y}_i$, (note : on a $\alpha_1 \neq 0$)

$$\begin{aligned}
\mathbf{A}^k\mathbf{x}_0 &= \sum_{i=1}^n \alpha_i \mathbf{A}^k \mathbf{y}_i = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{y}_i, \\
&= \alpha_1 \lambda_1^k \mathbf{y}_1 + \sum_{i=2}^n \alpha_i \lambda_i^k \mathbf{y}_i, \\
&= \alpha_1 \lambda_1^k \left(\mathbf{y}_1 + \sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{y}_i \right),
\end{aligned}$$

d'où

$$\lim_{k\to\infty} \left(\frac{|\lambda_1|}{\lambda_1}\right)^k \frac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|} = \frac{\alpha_1}{|\alpha_1|}\mathbf{y}_1.$$

Démonstration : on peut tout d'abord montrer par récurrence que $\mathbf{x}_k = \dfrac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|}$, puis en écrivant $\mathbf{x}_0 = \sum_{i=1}^n \alpha_i \mathbf{y}_i$, (note : on a $\alpha_1 \neq 0$)

$$\begin{aligned}\mathbf{A}^k\mathbf{x}_0 &= \sum_{i=1}^n \alpha_i \mathbf{A}^k \mathbf{y}_i = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{y}_i,\\ &= \alpha_1\lambda_1^k \mathbf{y}_1 + \sum_{i=2}^n \alpha_i\lambda_i^k \mathbf{y}_i,\\ &= \alpha_1\lambda_1^k\left(\mathbf{y}_1 + \sum_{i=2}^n \frac{\alpha_i}{\alpha_1}\left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{y}_i\right),\end{aligned}$$

d'où

$$\lim_{k\to\infty}\left(\frac{|\lambda_1|}{\lambda_1}\right)^k \frac{\mathbf{A}^k\mathbf{x}_0}{\|\mathbf{A}^k\mathbf{x}_0\|} = \frac{\alpha_1}{|\alpha_1|}\mathbf{y}_1.$$

Remarque : si on utilise la norme Euclidienne dans l'algorithme, alors

$$\lim_{k\to\infty} \mathbf{x}_k^\top \mathbf{A}\mathbf{x}_k = \mathbf{y}_1^\top \mathbf{A}\mathbf{y}_1 = \lambda_1 \mathbf{y}_1^\top \mathbf{y}_1 = \lambda_1 \|\mathbf{y}_1\|^2 = \lambda_1,$$

on peut ainsi fixer une tolérance *TOL* et utiliser le test

$$\|\mathbf{A}\mathbf{x}_k - (\mathbf{x}_k^\top \mathbf{A}\mathbf{x}_k)\mathbf{x}_k\| < TOL$$

pour terminer les itérations de l'algorithme.

Remarque : si on utilise la norme Euclidienne dans l'algorithme, alors

$$\lim_{k\to\infty} \mathbf{x}_k^\top \mathbf{A}\mathbf{x}_k = \mathbf{y}_1^\top \mathbf{A}\mathbf{y}_1 = \lambda_1 \mathbf{y}_1^\top \mathbf{y}_1 = \lambda_1 \|\mathbf{y}_1\|^2 = \lambda_1,$$

on peut ainsi fixer une tolérance *TOL* et utiliser le test

$$\|\mathbf{A}\mathbf{x}_k - (\mathbf{x}_k^\top \mathbf{A}\mathbf{x}_k)\mathbf{x}_k\| < TOL$$

pour terminer les itérations de l'algorithme.

Démo Scilab

Remarque : supposons **A** inversible. Si λ est valeur propre de **A** et **y** un vecteur propre associé alors

$$\mathbf{A}\mathbf{y} = \lambda\mathbf{y} \Leftrightarrow \mathbf{y} = \lambda\mathbf{A}^{-1}\mathbf{y} \Leftrightarrow \frac{1}{\lambda}\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}.$$

Donc si $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_{n-1}| > |\lambda_n|$ l'algorithme de la puissance appliqué à $\mathbf{A}^{-1}$ permet de déterminer $\frac{1}{\lambda_n}$ et $\mathbf{y}_n$.

Mais il n'est pas nécessaire d'inverser **A** !

Méthodes numériques

Méthode de la puissance inverse

Remarque : supposons **A** inversible. Si λ est valeur propre de **A** et **y** un vecteur propre associé alors

$$\mathbf{Ay} = \lambda \mathbf{y} \Leftrightarrow \mathbf{y} = \lambda \mathbf{A}^{-1}\mathbf{y} \Leftrightarrow \frac{1}{\lambda}\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}.$$

Donc si $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_{n-1}| > |\lambda_n|$ l'algorithme de la puissance appliqué à $\mathbf{A}^{-1}$ permet de déterminer $\frac{1}{\lambda_n}$ et $\mathbf{y}_n$.

Mais il n'est pas nécessaire d'inverser **A** !

Remarque : supposons **A** inversible. Si λ est valeur propre de **A** et **y** un vecteur propre associé alors

$$\mathbf{A}\mathbf{y} = \lambda\mathbf{y} \Leftrightarrow \mathbf{y} = \lambda\mathbf{A}^{-1}\mathbf{y} \Leftrightarrow \frac{1}{\lambda}\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}.$$

Donc si $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_{n-1}| > |\lambda_n|$ l'algorithme de la puissance appliqué à $\mathbf{A}^{-1}$ permet de déterminer $\frac{1}{\lambda_n}$ et $\mathbf{y}_n$.

Mais il n'est pas nécessaire d'inverser **A** !

Remarque : supposons $\mathbf{A}$ inversible. Si λ est valeur propre de $\mathbf{A}$ et $\mathbf{y}$ un vecteur propre associé alors

$$\mathbf{A}\mathbf{y} = \lambda\mathbf{y} \Leftrightarrow \mathbf{y} = \lambda\mathbf{A}^{-1}\mathbf{y} \Leftrightarrow \frac{1}{\lambda}\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}.$$

Donc si $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_{n-1}| > |\lambda_n|$ l'algorithme de la puissance appliqué à $\mathbf{A}^{-1}$ permet de déterminer $\frac{1}{\lambda_n}$ et $\mathbf{y}_n$.

Mais il n'est pas nécessaire d'inverser $\mathbf{A}$!

Remarque : supposons $\mathbf{A}$ inversible. Si λ est valeur propre de $\mathbf{A}$ et $\mathbf{y}$ un vecteur propre associé alors

$$\mathbf{Ay} = \lambda\mathbf{y} \Leftrightarrow \mathbf{y} = \lambda\mathbf{A}^{-1}\mathbf{y} \Leftrightarrow \frac{1}{\lambda}\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}.$$

Donc si $|\lambda_1| \geq |\lambda_2| \geq \ldots |\lambda_{n-1}| > |\lambda_n|$ l'algorithme de la puissance appliqué à $\mathbf{A}^{-1}$ permet de déterminer $\frac{1}{\lambda_n}$ et $\mathbf{y}_n$.

Mais il n'est pas nécessaire d'inverser $\mathbf{A}$!

Théorème

Soit $\mathbf{x}_0 \notin \mathrm{Vect}(\mathbf{y}_1, \ldots, \mathbf{y}_{n-1})$. Si on note $\mathbf{z}_k$ la solution du système linéaire $\mathbf{A}\mathbf{z}_k = \mathbf{x}_k$, la suite $(\mathbf{x}_k)$ définie par

$$\mathbf{x}_{k+1} = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|},$$

converge dans le sous-espace propre associé à $\mathbf{y}_n$ au sens où

$$\lim_{k\to\infty} \mathrm{signe}(\lambda_n)^k \mathbf{x}_k = \pm\mathbf{y}_n.$$

Remarque : comme les valeurs propres de $\mathbf{A} - q\mathbf{I}$ sont égales à $(\lambda_i - q)_{i=1\ldots n}$, l'algorithme de la puissance inverse appliqué à $\mathbf{A} - q\mathbf{I}$ permet de déterminer la valeur propre la plus proche de q.

Théorème

Soit $\mathbf{x}_0 \notin \mathrm{Vect}(\mathbf{y}_1, \ldots, \mathbf{y}_{n-1})$. Si on note $\mathbf{z}_k$ la solution du système linéaire $\mathbf{A}\mathbf{z}_k = \mathbf{x}_k$, la suite $(\mathbf{x}_k)$ définie par

$$\mathbf{x}_{k+1} = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|},$$

converge dans le sous-espace propre associé à $\mathbf{y}_n$ au sens où

$$\lim_{k\to\infty} \mathrm{signe}(\lambda_n)^k \mathbf{x}_k = \pm \mathbf{y}_n.$$

Remarque : comme les valeurs propres de $\mathbf{A} - q\mathbf{I}$ sont égales à $(\lambda_i - q)_{i=1\ldots n}$, l'algorithme de la puissance inverse appliqué à $\mathbf{A} - q\mathbf{I}$ permet de déterminer la valeur propre la plus proche de q.

Algorithme PageRank

Principe

L'algorithme PageRank permet d'attribuer une note (rank) à un ensemble de pages web.

Le moteur de recherche en ligne de Google permet, parmi des pages contenant un ensemble de mot-clés donné, de les présenter classées par ordre décroissant de cette note.

Article original de Lawrence Page et Sergey Brin :

`http://infolab.stanford.edu/pub/papers/google.pdf`

Algorithme PageRank

Principe

L'algorithme PageRank permet d'attribuer une note (rank) à un ensemble de pages web.

Le moteur de recherche en ligne de Google permet, parmi des pages contenant un ensemble de mot-clés donné, de les présenter classées par ordre décroissant de cette note.

Article original de Lawrence Page et Sergey Brin :

http://infolab.stanford.edu/pub/papers/google.pdf

Algorithme PageRank
Principe

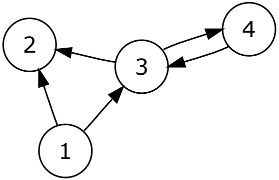
L'algorithme PageRank permet d'attribuer une note (rank) à un ensemble de pages web.

Le moteur de recherche en ligne de Google permet, parmi des pages contenant un ensemble de mot-clés donné, de les présenter classées par ordre décroissant de cette note.

Article original de Lawrence Page et Sergey Brin :

`http://infolab.stanford.edu/pub/papers/google.pdf`

Algorithme PageRank
Modélisation, graphe du web



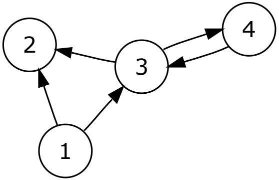
Représentons un ensemble de pages web $E = \{1, \ldots, n\}$ et les liens les associant $A \subset \{(x, y) \in E^2,\ x \neq y\}$ sous la forme d'un graphe orienté (E, A).

Matrice d'adjacence : $\mathbf{C}$ définie par $c_{ij} = 1$ s'il existe un lien entre les pages i et j. Ici on a

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Algorithme PageRank

Modélisation, graphe du web



Représentons un ensemble de pages web $E = \{1, \dots, n\}$ et les liens les associant $A \subset \{(x, y) \in E^2,\ x \neq y\}$ sous la forme d'un graphe orienté (E, A).

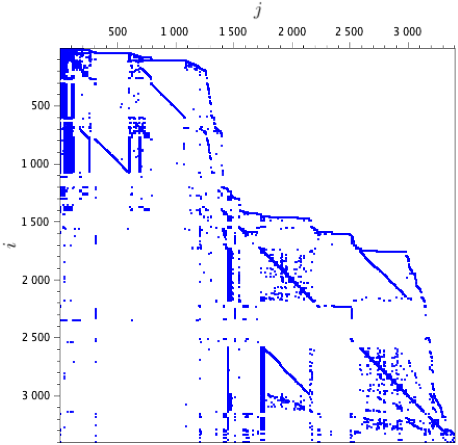
Matrice d'adjacence : $\mathbf{C}$ définie par $c_{ij} = 1$ s'il existe un lien entre les pages i et j. Ici on a

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Algorithme PageRank

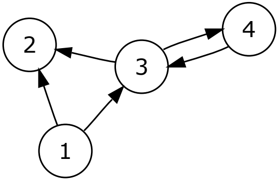
Modélisation, graphe du web

Voici par exemple la matrice **C** des pages du domaine `utc.fr` accessibles à partir de `https://ww.utc.fr`, représentée dans Scilab (un point bleu est un élément non-nul) :



Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

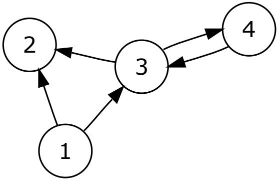
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

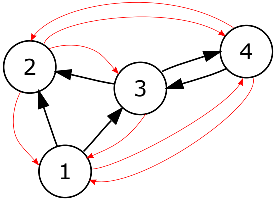
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

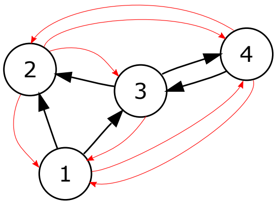
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

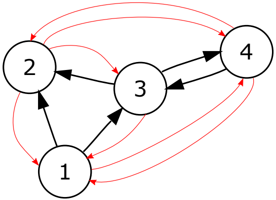
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

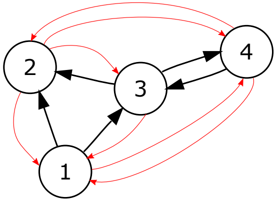
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

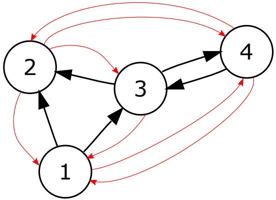
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

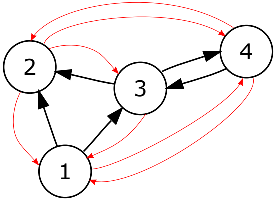
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

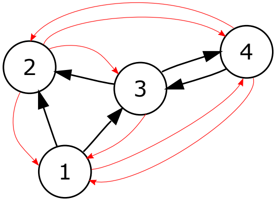
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q \frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Modèle d'un promeneur du web : à partir d'une page i (typiquement $q = 0,85$)

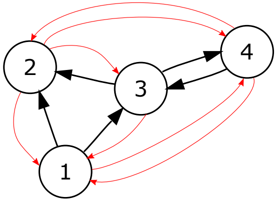
- avec la probabilité $1 - q$: on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{1-q}{n}$
- avec la probabilité q :
  - si la page n'a pas de lien, on choisit aléatoirement une des n pages $\rightarrow$ *proba* $= \dfrac{q}{n}$
  - sinon on suit aléatoirement l'un des $\sum_{j=1}^{n} c_{ij}$ liens $\rightarrow$ *proba* $= \dfrac{q}{\sum_{j=1}^{n} c_{ij}}$

La probabilité de passer de la page i à la page j est donc égale à

$$p_{ij} = \begin{cases} \frac{1}{n}, & \text{si } \sum_{j=1}^{n} c_{ij} = 0, \\ q \frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} + \frac{1-q}{n}, & \text{sinon.} \end{cases}$$

Algorithme PageRank

Modélisation, promeneur du web



Par construction la matrice $\mathbf{P}$ est stochastique : on a $\sum_j p_{ij} = 1$. Dans notre exemple on a

$$\mathbf{P} = q \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 \end{pmatrix} + \frac{1-q}{4}\mathbf{e}\mathbf{e}^\top,$$

où $\mathbf{e}^\top = (1, 1, 1, 1)$.

Définition

Soit $E = \{1, \ldots, n\}$. On appelle processus de Markov la suite de variables aléatoires $(X_k)_{k\in\mathbb{N}}$ prenant leurs valeurs dans E et définies par

$$\forall n \geq 0,\ \forall (i,j) \in E^2,\ \mathrm{Prob}(X_{n+1} = j \,|\, X_n = i) = p_{ij},$$

où $\mathbf{P}$ est une matrice stochastique, et par la distribution initiale

$$\pi_0 = (\mathrm{Prob}(X_0 = 1), \ldots, \mathrm{Prob}(X_0 = n))\,.$$

Remarque : on peut imposer $X_0 = i$ en prenant

$$\pi_0 = (0, \ldots, 0, 1, 0, \ldots, 0).$$

Définition

Soit $E = \{1, \ldots, n\}$. On appelle processus de Markov la suite de variables aléatoires $(X_k)_{k \in \mathbb{N}}$ prenant leurs valeurs dans E et définies par

$$\forall n \geq 0,\ \forall (i,j) \in E^2,\ \mathrm{Prob}(X_{n+1} = j \,|\, X_n = i) = p_{ij},$$

où $\mathbf{P}$ est une matrice stochastique, et par la distribution initiale

$$\pi_0 = (\mathrm{Prob}(X_0 = 1), \ldots, \mathrm{Prob}(X_0 = n)) .$$

Remarque : on peut imposer $X_0 = i$ en prenant

$$\pi_0 = (0, \ldots, 0, 1, 0, \ldots, 0).$$

Propriété

La distribution du processus de Markov au temps k définie par le vecteur ligne

$$\pi_k = (\mathrm{Prob}(X_k = 1), \ldots, \mathrm{Prob}(X_k = n))$$

vérifie $\pi_{k+1} = \pi_k \mathbf{P}$.

Démonstration : D'après la formule des probabilités totales on a

$$\begin{aligned}(\pi_{k+1})_j = \mathrm{Prob}(X_{k+1} = j) &= \textstyle\sum_{i=1}^n \mathrm{Prob}(X_{k+1} = j \,|\, X_k = i)\, \mathrm{Prob}(X_k = i),\\ &= \textstyle\sum_{i=1}^n p_{ij} (\pi_k)_i.\end{aligned}$$

Propriété

La distribution du processus de Markov au temps k définie par le vecteur ligne

$$\pi_k = (\mathrm{Prob}(X_k = 1), \ldots, \mathrm{Prob}(X_k = n))$$

vérifie $\pi_{k+1} = \pi_k \mathbf{P}$.

Démonstration : D'après la formule des probabilités totales on a

$$\begin{aligned}(\pi_{k+1})_j = \mathrm{Prob}(X_{k+1} = j) &= \textstyle\sum_{i=1}^n \mathrm{Prob}(X_{k+1} = j \,|\, X_k = i)\,\mathrm{Prob}(X_k = i),\\ &= \textstyle\sum_{i=1}^n p_{ij} (\pi_k)_i.\end{aligned}$$

Propriété

La distribution du processus de Markov au temps k définie par le vecteur ligne

$$\pi_k = (\mathrm{Prob}(X_k = 1), \ldots, \mathrm{Prob}(X_k = n))$$

vérifie $\pi_{k+1} = \pi_k \mathbf{P}$.

Démonstration : D'après la formule des probabilités totales on a

$$(\pi_{k+1})_j = \mathrm{Prob}(X_{k+1} = j) = \sum_{i=1}^n \mathrm{Prob}(X_{k+1} = j \mid X_k = i)\,\mathrm{Prob}(X_k = i),$$
$$= \sum_{i=1}^n p_{ij}(\pi_k)_i.$$

Propriété

La distribution du processus de Markov au temps k définie par le vecteur ligne

$$\pi_k = (\mathrm{Prob}(X_k = 1), \ldots, \mathrm{Prob}(X_k = n))$$

vérifie $\pi_{k+1} = \pi_k \mathbf{P}$.

Démonstration : D'après la formule des probabilités totales on a

$$\begin{aligned}(\pi_{k+1})_j = \mathrm{Prob}(X_{k+1} = j) &= \textstyle\sum_{i=1}^n \mathrm{Prob}(X_{k+1} = j \,|\, X_k = i)\, \mathrm{Prob}(X_k = i),\\ &= \textstyle\sum_{i=1}^n p_{ij} (\pi_k)_i.\end{aligned}$$

La distribution stationnaire $\hat{\pi} = \lim_{k\to\infty} \pi_k$, si celle-ci existe, définit le classement de l'algorithme PageRank. Puisque par définition $\pi_{k+1} = \pi_k \mathbf{P}$, cette distribution vérifie nécessairement $\hat{\pi}\mathbf{P} = \hat{\pi}$.

Théorème

Soit $\mathbf{P}$ une matrice stochastique irréductible (vérifiant $p_{ij} > 0\ \forall i,j$). Alors $\mathbf{P}$ admet $\lambda = 1$ pour valeur propre dominante unique et il existe un vecteur propre à gauche associé $\hat{\pi}$ vérifiant

$$\hat{\pi}_i \geq 0,\ i = 1 \dots n \qquad \text{et} \qquad \sum_{i=1}^{n} \hat{\pi}_i = 1.$$

Conclusion : la formule $\pi_{k+1} = \pi_k \mathbf{P}$ est l'itération de la méthode de la puissance pour la norme $\|\mathbf{x}\| = \sum_{i=1}^{n} |x_i|$ et le théorème garantit qu'elle converge !

Algorithme PageRank

Théorème de Perron-Frobenius

La distribution stationnaire $\hat{\pi} = \lim_{k\to\infty} \pi_k$, si celle-ci existe, définit le classement de l'algorithme PageRank. Puisque par définition $\pi_{k+1} = \pi_k \mathbf{P}$, cette distribution vérifie nécessairement $\hat{\pi}\mathbf{P} = \hat{\pi}$.

Théorème

Soit **P** une matrice stochastique irréductible (vérifiant $p_{ij} > 0\ \forall i, j$). Alors **P** admet $\lambda = 1$ pour valeur propre dominante unique et il existe un vecteur propre à gauche associé $\hat{\pi}$ vérifiant

$$\hat{\pi}_i \geq 0,\ i = 1 \ldots n \qquad \text{et} \qquad \sum_{j=1}^{n} \hat{\pi}_i = 1.$$

Conclusion : la formule $\pi_{k+1} = \pi_k \mathbf{P}$ est l'itération de la méthode de la puissance pour la norme $\|\mathbf{x}\| = \sum_{i=1}^{n} |x_i|$ et le théorème garantit qu'elle converge !

La distribution stationnaire $\hat{\pi} = \lim_{k\to\infty} \pi_k$, si celle-ci existe, définit le classement de l'algorithme PageRank. Puisque par définition $\pi_{k+1} = \pi_k \mathbf{P}$, cette distribution vérifie nécessairement $\hat{\pi}\mathbf{P} = \hat{\pi}$.

Théorème

Soit **P** une matrice stochastique irréductible (vérifiant $p_{ij} > 0 \ \forall i, j$). Alors **P** admet $\lambda = 1$ pour valeur propre dominante unique et il existe un vecteur propre à gauche associé $\hat{\pi}$ vérifiant

$$\hat{\pi}_i \geq 0,\ i = 1 \ldots n \qquad \text{et} \qquad \sum_{j=1}^{n} \hat{\pi}_i = 1.$$

Conclusion : la formule $\pi_{k+1} = \pi_k \mathbf{P}$ est l'itération de la méthode de la puissance pour la norme $\|\mathbf{x}\| = \sum_{i=1}^{n} |x_i|$ et le théorème garantit qu'elle converge !

Algorithme PageRank

En pratique

On ne forme pas la matrice $\mathbf{P}$ et on n'effectue pas non plus le produit $\pi_k \mathbf{P}$ car les itérations de l'algorithme de la puissance reviennent à parcourir une base de données des pages en mettant à jour π_k.

En TD avec Scilab nous adopterons une approche intermédiaire en faisant effectivement le produit mais en utilisant des matrices creuses (on ne stocke en mémoire que les élément non nuls).