Algoritmy pro rozsáhlá data

Pagerank

Jan Konecny

10. prosince 2021

Příklad 1

Mějme

$$A = \begin{pmatrix} 0 & 5 & -10 \\ 0 & 22 & 16 \\ 0 & -9 & -2 \end{pmatrix} \qquad \vec{x}_1 = \begin{pmatrix} -5 \\ -4 \\ 3 \end{pmatrix} \qquad \vec{x}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$A\vec{x}_1 = \begin{pmatrix} -50 \\ -40 \\ 30 \end{pmatrix} = -10 \cdot \begin{pmatrix} -5 \\ -4 \\ 3 \end{pmatrix} \qquad A\vec{x}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 0 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

- Pro tyto vektory se násobení maticí A rovná násobení nějakým skalárem (konkrétně $\lambda_1 = 10$, $\lambda_2 = 0$)
- Jinak řečeno, násobením maticí A dostaneme vektor se stejným směrem.

Definice 2

Nechť A je $n \times n$ matice, $x \in \mathbb{C}^n$ je nenulový vektor, t.ž.

$$A\vec{x} = \lambda\vec{x}$$

pro nějaký skalár λ.
Pak λ nazýváme vlastní hodnota matice A, a $\vec{x}$ se nazývá vlastní vektor matice A vzhledem k λ.

Uvažujme, že nenulový vektor $\vec{x}$ splňuje $A\vec{x} = \lambda\vec{x}$.
Pak

$$A\vec{x} - \lambda\vec{x} = \vec{o} \quad \text{neboli} \quad (A - \lambda I)\vec{x} = \vec{o} \quad \text{neboli} \quad (\lambda I - A)\vec{x} = \vec{o}.$$

Tedy když hledáme vlastní vektory, hledáme netriviální řešení tohoto homogenního systému rovnic.

Připomeňme si, že homogenní systém rovnic sestává:

- ze základních řešení
- a z lineárních kombinací těchto základních řešení.

Základní řešení rovnice $(\lambda I - A)\vec{x} = \vec{o}$ nazýváme základní vlastní vektory.

Uvažujme, že $(\lambda I - A)$ je invertibilní (tj. existuje inverze $(\lambda I - A)^{-1}$. Pak

$$\begin{aligned}
\vec{x} &= I\vec{x} \\
&= \left((\lambda I - A)^{-1}(\lambda I - A)\right)\vec{x} \\
&= (\lambda I - A)^{-1}\left((\lambda I - A)\vec{x}\right) \\
&= (\lambda I - A)^{-1}\vec{o} \\
&= \vec{o}
\end{aligned}$$

To je ale spor s předpokladem, takže $(\lambda I - A)$ nemůže mít inverzi.

Pokud matice nemá inverzi, pak je její determinant roven 0.
A tedy

$$\det(\lambda I - A) = 0 \quad \text{neboli} \quad \det(A - \lambda I) = 0$$

- Výraz $\det(\lambda I - A)$ je polynom (s proměnnou x) nazývaný charakteristický polynom A.
- $\det(\lambda I - A) = 0$ se nazývá charakteristická rovnice.
- Vlastním hodnotám se proto taky někdy říká charakteristické hodnoty.

Následující věta říká, že kořeny charakteristického polynomu jsou vlastní vlastní hodnoty A.

Věta 3

Nechť A je $n \times n$ matice a platí

$$\det(\lambda I - A) = 0$$

pro nějaké $\lambda \in \mathbb{C}$.
Pak λ je vlastní hodnota A a tedy existuje nenulový vektor $\vec{x} \in \mathbb{C}^n$ t.ž. $A\vec{x} = \lambda\vec{x}$.

Definice 4

Nechť A a B jsou $n \times n$ matice, t.ž. existuje invertibilní matice P, tak že

$$A = P^{-1}BP.$$

Pak se

- A a B nazývají podobné matice;
- P se nazývá podobnostní transformace.

- Pojem podobných matic můžeme využít pro hledání vlastních hodnot:

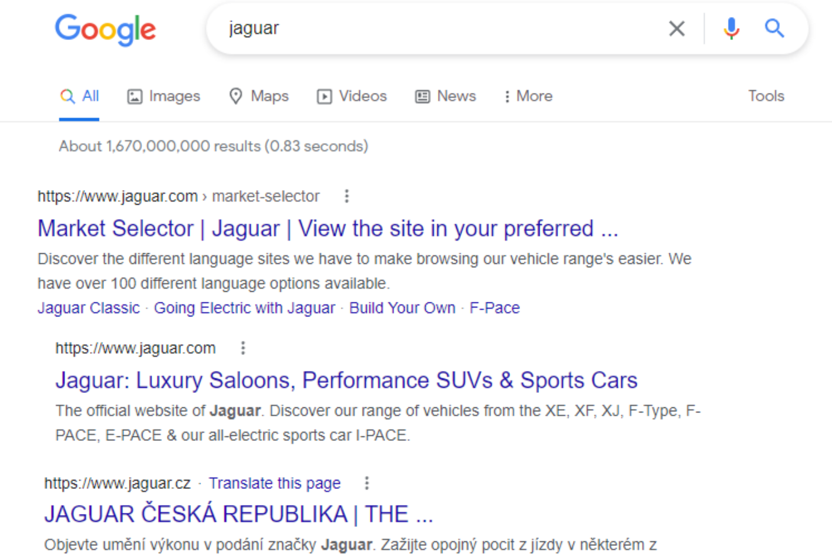
Lemma 5

Podobné matice mají stejné vlastní hodnoty.

Poznámka

Vlastní vektory podobných matic jsou obecně *různé*.

PageRank


Google
jaguar
All
Images
Maps
Videos
News
More
Tools
About 1,670,000,000 results (0.83 seconds)
https://www.jaguar.com › market-selector
Market Selector | Jaguar | View the site in your preferred ...
Discover the different language sites we have to make browsing our vehicle range's easier. We have over 100 different language options available.
Jaguar Classic · Going Electric with Jaguar · Build Your Own · F-Pace
https://www.jaguar.com
Jaguar: Luxury Saloons, Performance SUVs & Sports Cars
The official website of Jaguar. Discover our range of vehicles from the XE, XF, XJ, F-Type, F-PACE, E-PACE & our all-electric sports car I-PACE.
https://www.jaguar.cz · Translate this page
JAGUAR ČESKÁ REPUBLIKA | THE ...
Objevte umění výkonu v podání značky Jaguar. Zažijte opojný pocit z jízdy v některém z

PageRank

PageRank

- je pojmenován po termínu „web page“ a současně pro spoluzakladateli – Larry Page.
- je algoritmus, který Google Search používá k hodnocení webových stránek ve výsledcích jejich vyhledávačů.
- způsob měření důležitosti webových stránek.

Podle Google:

> *PageRank funguje tak, že počítá počet a kvalitu odkazů na stránku a určuje hrubý odhad důležitosti webu. Základním předpokladem je, že důležitější webové stránky pravděpodobně obdrží více odkazů z jiných webových stránek.*

Uvažujme množinu webových stránek uspořádanou od 1 do n, a i jako určitou webovou stránku.

- outlink stránky i – webová stránka, na kterou se odkazuje i
- O_i – množina všech outlinků stránky i
- N_i – počet všech outlinků stránky i, tj. $|O_i| = N_i$
- inlink stránky i – webová stránka, která se odkazuje na i
- I_i – množina všech inlinků stránky i

obecné pozorování: Stránka je obecně považovaná za tím důležitější, čím více má inlinků.

problém: Hodnocení založené čistě na počtu inlinků se dá snadno zmanipulovat:
např. můžeme uměle zvyšovat důležitost stránky tak, že vytvoříme velké množství stránek s outlinky na i

řešení: Aby jsme efekt té manipulace minimalizoval, započteme do hodnocení (ranku) stránky taky rank jejích inlinků:

- rank stránky i je vážená suma ranků těch stránek, které mají outlinky na i
- vážení je uděláno tak, že se rank stránky i rovnoměrně rozdělí mezi všechny její outlinky.

Přepsáno do matematického vzorce, dostáváme

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}$$

- Kvůli rekurenci to nemůže být vypočteno přímo.
- K výpočtu se používá iterační metoda.

- K výpočtu se používá iterační metoda:

Odhadne se počáteční vektor $\vec{r}^{(0)}$ ranků a pak se iteruje

$$r_i^{(k+1)} = \sum_{j \in I_i} \frac{r_j^{(k)}}{N_j}, \qquad k = 0, 1, 2, \ldots \tag{1}$$

dokud nedostaneme konvergenci, tj, dokud nenastane

$$\vec{r}^{(k)} \approx \vec{r}^{(k+1)} \qquad \text{pro nějaké } k.$$

Problém: Pokud nějaká stránka nemá žádné outlinky…

Pro lepší vhled si přeformulujeme (1) na násobení matice a vektoru.

Pro lepší vhled si přeformulujeme (1) na násobení matice a vektoru:

Nechť Q je čtvercová matice dimenze n, t.ž.

$$Q_{ij} = \begin{cases} \frac{1}{N_j} & \text{pokud existuje link z } j \text{ na } i, \\ 0 & \text{jinak.} \end{cases}$$

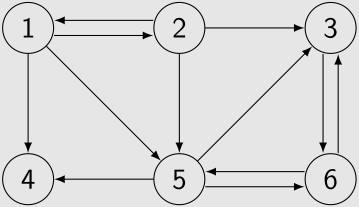
To znamená, že

- nenulové hodnoty v řádku i odpovídají inlinkům stránky i
- nenulové hodnoty ve sloupci j odpovídají outlinkům stránky j
- tyto hodnoty jsou $\frac{1}{N_j}$ a součty hodnot ve sloupci je 1 (pokud není celý sloupec nulový).

Nechť Q je čtvercová matice dimenze n, t.ž.

$$Q_{ij} = \begin{cases} \frac{1}{N_j} & \text{pokud existuje link z } j \text{ na } i, \\ 0 & \text{jinak.} \end{cases}$$

Příklad 6



$$\Longrightarrow \qquad Q = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

$$r_i^{(k+1)} = \sum_{j \in I_i} \frac{r_j^{(k)}}{N_j} \quad \text{můžeme zapsat jako} \quad \vec{r}^{(k+1)} = Q\vec{r}^{(k)}$$

$$r_i^{(k+1)} = \sum_{j \in I_i} \frac{r_j^{(k)}}{N_j} \quad \text{můžeme zapsat jako} \quad \vec{r}^{(k+1)} = Q\vec{r}^{(k)}$$

Když ještě uvážíme podmínku konvergence

$$\vec{r}^{(k)} \approx \vec{r}^{(k+1)} \qquad \text{pro nějaké } k.$$

Dostaneme, že vlastně hledáme vektor $\vec{r}$ tak, aby:

$$Q\vec{r} = \vec{r}$$

neboli

$$Q\vec{r} = \lambda\vec{r} \qquad \text{pro } \lambda = 1.$$

Problém: Pokud nějaká stránka nemá žádné outlinky...

Příklad 7

$$Q = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{3} & 0 \end{pmatrix} \qquad \vec{r}^{(0)} = \begin{pmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \end{pmatrix}$$

$$\vec{r}^{(1)} = \begin{pmatrix} \frac{1}{18} \\ \frac{1}{18} \\ \frac{7}{36} \\ \frac{2}{18} \\ \frac{7}{36} \\ \frac{2}{9} \end{pmatrix} \approx \begin{pmatrix} 0.06 \\ 0.06 \\ 0.19 \\ 0.11 \\ 0.19 \\ 0.22 \end{pmatrix} \quad \vec{r}^{(2)} = \begin{pmatrix} 0.02 \\ 0.02 \\ 0.19 \\ 0.08 \\ 0.15 \\ 0.26 \end{pmatrix} \quad \vec{r}^{(3)} = \begin{pmatrix} 0.01 \\ 0.01 \\ 0.19 \\ 0.06 \\ 0.14 \\ 0.24 \end{pmatrix}$$

Co se tam vlastně děje špatně:

- 4-tý sloupec Q je nulový.
- cokoli je v $\vec{r}^{(k)}$ na 4-tém řádku je násobeno 0.
- zatímco ranky na ostatních řádcích se „rozlijí“ do ranků svých outlinků, ranky na 4-tém řádku zmizí (ale neustále přitéká).
- toto nutně konverguje k 0 (zde součet ranků v $\vec{r}^{(100)}$ je 0.0002).

Aby nám rank neodtékal:

- ze stránek bez outlinků uděláme stránky, které mají outlinky na všechny stránky.

Definujeme vektor $\vec{d}$ jako

$$d_j = \begin{cases} 1 & \text{pokud } N_j = 0 \\ 0 & \text{jinak} \end{cases}$$

pro $j = 1, \ldots, n$ a

$$\vec{e} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n$$

Modifikovaná matice je pak dána takto

$$P = Q + \frac{1}{n}\vec{e}\vec{d}^{\mathrm{T}}.$$

$$P = Q + \frac{1}{n}\vec{e}\vec{d}^{\mathrm{T}}.$$

Takováto matice P je sloupcově stochastická, tj.

- má nezáporné prvky,
- prvky v každém sloupci dávají součet 1.

Ekvivalentně se dá charakterizovat takto:

Věta 8

Sloupcově stochastická matice P splňuje

$$\vec{e}^{\mathrm{T}} P = \vec{e}^{\mathrm{T}},$$

kde $\vec{e}$ je definováno

$$\vec{e} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n.$$

Příklad 9

Matice z předchozích příkladů

$$Q = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

bud modifikována na následující matici:

$$P = \begin{pmatrix} 0 & \frac{1}{3} & 0 & \frac{1}{6} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{6} & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{6} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{6} & \frac{1}{3} & 0 \end{pmatrix}$$

Analogicky k

$$Q\vec{r} = \lambda\vec{r} \qquad \lambda = 1$$

bychom měli definovat rank jako vlastní vektor matice P příslušný k vlastní hodnotě $\lambda = 1$:

$$P\vec{r} = \lambda\vec{r} \qquad \lambda = 1$$

Ale:

- nemáme zajištěnou unikátnost vlastního vektoru s vlastní hodnotou $\lambda = 1$;
- unikátnost nám zajistí ireducibilita matice...

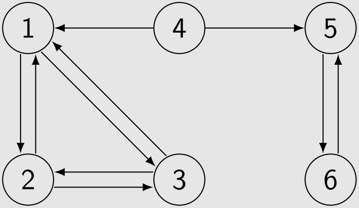
Definice 10

Čtvercová matice se nazývá reducibilní, pokud existuje permutační matice P (tj. matice s jednou jedničkou v každém řádku a v každém sloupci, jinak nuly), t.ž.

$$PAP^{\mathrm{T}} = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix},$$

kde X a Z jsou čtvercové matice.
V opačném případě se matice nazývá ireducibilní.

Příklad 11



$$\Longrightarrow \qquad Q = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- Graf odpovídající ireducibilní matici je silně souvislý:
  tj. mezi každými dvěma uzly existuje orientovaná cesta.
- Následující věta nám pak zaručí existenci vlastního vektoru.

Značení:

- $A > 0$ označuje, že A je striktně pozitivní ($A_{ij} > 0$ pro všechna i, j).
- λ_1 označuje dominantní vlastní hodnotu – tj. největší ze všech vlastních hodnot.

Věta 12 (Perron-Frobeniova věta)

Nechť A je ireducibilní sloupcově stochastická matice, pak

- $\lambda_1 = 1$;
- *existuje unikátní odpovídající vlastní vektor* $\vec{r}$ *splňující* $\vec{r} > 0$ *a* $\|\vec{r}\| = 1$;
- *toto je jediný vlastní vektor, který je nezáporný;*
- *pokud* $A > 0$, *pak také platí* $|\lambda_i| < 1, \quad i = 2, 3, \ldots, n$.

- Vzhledem k velikosti Internetu (popř. počtu indexovaných stránek) můžeme s jistotou považovat matici P za reducibilní.
- Takže PageRankový vlastní vektor není dobře definovaný.
- Abychom zajistili ireducibilitu, přidáme link z každé stránky na všechny ostatní – tento přidaný link ale bude bude mít nižší hodnotu:

$$A = \alpha P + (1-\alpha)\frac{1}{n}\vec{e}\vec{e}^{\mathrm{T}}$$

pro nějaké $a \in \langle 0, 1 \rangle$.

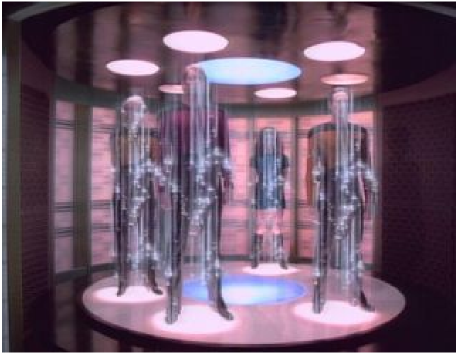
- Zjevně taková matice je sloupcově stochastická:

$$\vec{e}^{\mathrm{T}} A = \alpha \vec{e}^{\mathrm{T}} P + (1-\alpha)\frac{1}{n}\vec{e}^{\mathrm{T}}\vec{e}\vec{e}^{\mathrm{T}} = \alpha \vec{e}^{\mathrm{T}} + (1-\alpha)\vec{e}^{\mathrm{T}} = \vec{e}^{\mathrm{T}}.$$

Trochu jiný pohled na PageRank

- Máme „náhodného webového surfaře" – sedí u prohlížeče a náhodně kliká na linky.
- PageRankový vektor r pak odpovídá rozdělení pravděpodobnosti s jakou bude na jednotlivých stránkách po dostatečně dlouhé době.
- To, že se nezastaví (vždy bude na co klikat) jsme už zajistili doplněním matice Q na sloupcově stochastickou matici P.
- Pokud by se nešlo dostat z každé stránky na každou (graf by nebyl silně souvislý) $\vec{r}$ by bylo závislé na výchozí pozici.
- Použitím matice A vlastně říkáme, že s jakousi malou pravděpodobností $(1-\alpha)$ skočí na libovolnou náhodnou (indexovanou) stránku.

- Použitím matice A vlastně říkáme, že s jakousi malou pravděpodobností $(1 - \alpha)$ skočí na libovolnou náhodnou (indexovanou) stránku.
- Tomu se v kontextu náhodného surfaře říká teleportace.



- Teď ukážeme, že PageRank matice A je dobře definovaný

Věta 13

Sloupcově stochastická matice A definovaná v

$$A = \alpha P + (1-\alpha)\frac{1}{n}\vec{e}\vec{e}^{\mathrm{T}}$$

je ireducibilní (protože $A > 0$) a má dominantní hlavní hodnotu $\lambda_1 = 1$. Odpovídající vlastní vektor $\vec{r}$ splňuje $\vec{r} > 0$.

Důkaz.

Uvádím bez důkazu. $\square$

Kvůli konvergenci numerického algoritmu pro výpočet vlastního vektoru $\vec{r}$ je nutné prozkoumat, jak se mění vlastní hodnoty matice matice P po této modifikaci.

Věta 14

Uvažujme, že vlastní hodnoty matice P jsou $\{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$.
Pak vlastní hodnoty matice

$$A = \alpha P + (1-\alpha)\frac{1}{n}\vec{e}\vec{e}^{\mathrm{T}}$$

jsou $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_n\}$.

Důkaz.

Definujme $\hat{e}$ jako normalizovaný vektor $\vec{e}$ na Euklidovskou velikost 1, tj

$$\hat{e} = \left\langle n^{-\frac{1}{2}}, \ldots, n^{-\frac{1}{2}} \right\rangle$$

Důkaz.

Nechť $U_1 \in \mathbb{R}^{n\times(n-1)}$ je taková matice, že

$$U = \begin{pmatrix} \hat{e} & U_1 \end{pmatrix}$$

je ortogonální.
Protože platí $\hat{e}^{\mathrm{T}}P = \hat{e}^{\mathrm{T}}$, dostáváme

$$\begin{aligned} U^{\mathrm{T}}PU &= \begin{pmatrix} \hat{e}^{\mathrm{T}}P \\ U_1^{\mathrm{T}}P \end{pmatrix} \begin{pmatrix} \hat{e} & U_1 \end{pmatrix} = \begin{pmatrix} \hat{e}^{\mathrm{T}} \\ U_1^{\mathrm{T}}P \end{pmatrix} \begin{pmatrix} \hat{e} & U_1 \end{pmatrix} \\ &= \begin{pmatrix} \hat{e}^{\mathrm{T}}\hat{e} & \hat{e}^{\mathrm{T}}U_1 \\ U_1^{\mathrm{T}}P\hat{e} & U_1^{\mathrm{T}}P^{\mathrm{T}}U_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \vec{w} & T \end{pmatrix} \end{aligned}$$

kde $\vec{w} = U_1^{\mathrm{T}}P\hat{e}$ a $T = U_1^{\mathrm{T}}P^{\mathrm{T}}U_1$.

Důkaz.

Protože jsme provedli podobnostní transformací, matice T bude mít vlastní hodnoty $\lambda_2, \lambda_3, \ldots, \lambda_n$.

Dále máme

$$U^{\mathrm{T}}\vec{v} = \begin{pmatrix} n^{-\frac{1}{2}}\vec{e}^{\mathrm{T}}\vec{v} \\ U_1^{\mathrm{T}}\vec{v} \end{pmatrix} = \begin{pmatrix} n^{-\frac{1}{2}} \\ U_1^{\mathrm{T}}\vec{v} \end{pmatrix}$$

A tedy

$$\begin{aligned} U^{\mathrm{T}}AU &= U^{\mathrm{T}}(\alpha P + (1-\alpha)\vec{v}\vec{e}^{\mathrm{T}})U \\ &= \alpha\begin{pmatrix} 1 & 0 \\ \vec{w} & T \end{pmatrix} + (1-\alpha)\begin{pmatrix} n^{-\frac{1}{2}} \\ U_1^{\mathrm{T}}\vec{v} \end{pmatrix}\begin{pmatrix} n^{\frac{1}{2}} & 0 \end{pmatrix} \\ &= \alpha\begin{pmatrix} 1 & 0 \\ \vec{w} & T \end{pmatrix} + (1-\alpha)\begin{pmatrix} 1 & 0 \\ n^{\frac{1}{2}}U_1^{\mathrm{T}}\vec{v} & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \vec{w}_1 & \alpha T \end{pmatrix} \end{aligned}$$

Tvrzení pak z toho přímo vyplývá.

Ta věta vlastně říká, že i když P má více vlastních hodnot $\lambda_i = 1$ (což je případ Google matice), druhá největší hodnota matice A je vždy rovna α.

Příklad 15

Matice odpovídající předchozímu případu:

```
P = np.matrix([[ 0,.5,.5,.5, 0, 0],
               [.5, 0,.5, 0, 0, 0],
               [.5,.5, 0, 0, 0, 0],
               [ 0, 0, 0, 0, 0, 0],
               [ 0, 0, 0,.5, 0, 1],
               [ 0, 0, 0, 0, 1, 0]])

np.linalg.eig(P)[0]
Out: array([-0.5,  1. , -0.5,  1. , -1. ,  0. ])
```

Příklad 16

```
np.linalg.eig(P)[0]
Out: array([-0.5,  1. , -0.5,  1. , -1. ,  0. ])
```

```
np.linalg.eig(P)[1][:,1]
Out:
matrix([[0.57735027],
        [0.57735027],
        [0.57735027],
        [0.        ],
        [0.        ],
        [0.        ]])
```

```
np.linalg.eig(P)[1][:,3]
Out:
matrix([[0.        ],
        [0.        ],
        [0.        ],
        [0.        ],
        [0.70710678],
        [0.70710678]])
```

Příklad 17

```
alpha = .85
A = alpha * P + (1-alpha)* 1/6

np.linalg.eig(A)[0]
Out: array([ 1.00000e+00,  8.50000e-01, -8.06313e-17,
            -8.50000e-01, -4.25000e-01, -4.25000e-01])

r = np.linalg.eig(A)[1][:,0]
r = r/sum(r)

r
Out:
matrix([[0.19524854], [0.1877924 ], [0.1877924 ],
        [0.025     ], [0.20495495], [0.19921171]])
```

Příklad 18

```
r
Out:
matrix([[0.19524854], [0.1877924 ], [0.1877924 ],
        [0.025     ], [0.20495495], [0.19921171]])
```

```
A * r
Out:
matrix([[0.19524854], [0.1877924 ], [0.1877924 ],
        [0.025     ], [0.20495495], [0.19921171]])
```

Poznámka

Místo modifikace

$$A = \alpha P + (1-\alpha)\frac{1}{n}ee^T$$

můžeme použít

$$A = \alpha P + (1-\alpha)\frac{1}{n}ve^T$$

kde v je tzv. personalizační vektor

- je nezáporný vektor s $\|v\|_1 = 1$
- může být vybrán tak, aby ovlivňoval vyhledávání některých druhů webových stránek (link farmy).

- Chceme tedy řešit problém vlastní hodnoty

$$A\vec{r} = \vec{r},$$

kde $\vec{r}$ je normalizovaný – $\|\vec{r}\|_1 = 1$.

- Jediný schůdný algoritmus je mocninná metoda:

---

for $k = 1, 2, \ldots$ *until convergence* **do**
  $\vec{q}^{(k)} \leftarrow A\vec{r}^{(k-1)}$;
  $\vec{r}^{(k)} \leftarrow \vec{q}^{(k)} / \|\vec{q}^{(k)}\|_1$;

- $r^{(0)}$ je daná iniciální aproximace
- normalizace (spodní řádek) se dělá, aby se zabránilo situaci, kdy je vektor příliš velký či malý, a nedá se pak dobře reprezentovat v plovoucí řádové čárce.
- uvidíme později, že pro Pagerank ta normalizace není potřeba.

Konvergence metody záleží na rozložení vlastních hodnot.

- Pro zjednodušení budeme předpokládat, že A je diagonalizovatelná, tj. existuje regulární matice T z vlastních vektorů A, t.ž.

$$T^{-1}AT = \mathrm{diag}(\lambda_1, \dots, \lambda_n).$$

- Uspořádejme vlastní hodnoty λ_i jsou uspořádané

$$1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_n|$$

- Rozložíme iniciální aproximaci $\vec{r}^{(0)}$ podle vlastních vektorů

$$\vec{r}^{(0)} = c_1\vec{t}_1 + c_2\vec{t}_2 + \cdots + c_n\vec{t}_n,$$

kde se předpokládá, že $c_1 \neq 0$ a $\vec{r} = \vec{t}_1$ je hledaný vlastní vektor.

Pak máme

$$\begin{aligned} A^k r^{(0)} &= c_1 A^k \vec{t_1} + c_2 A^k \vec{t_2} + \cdots + c_n A^k \vec{t_n} \\ &= c_1 \lambda_1^k \vec{t_1} + c_2 \lambda_2^k \vec{t_2} + \cdots + c_n \lambda_n^k \vec{t_n} \\ &= c_1 \vec{t_1} + \sum_{j=2}^{n} c_j \lambda_j^k \vec{t_j} \end{aligned}$$

Zjevně $\sum_{j=2}^{n} c_j \lambda_j^k \vec{t_j}$ jde k nule pro rostoucí k, protože pro $j = 2, 3, \ldots$ máme $|\lambda_j| < 1$.
Rychlost konvergence je tedy určena hodnotou $|\lambda_2|$:

- když je $|\lambda_2|$ blízko 1, je konvergence pomalá;
- u Google matice je $|\lambda_2| = \alpha$.

K té normalizaci $r^{(k)} \leftarrow q^{(k)}/\|q^{(k)}\|_1$:

- V obvyklém použití mocninné metody je vektor normalizován, aby se zabránilo přetečení nebo podtečení.
- Ukážeme si, že pro sloupcově stochastickou matici to není třeba.

Věta 19

Předpokládejme, že vektor $\vec{z}$ splňuje $\|\vec{z}\|_1 = \vec{e}^{\mathrm{T}}\vec{z} = 1$ a že matice A je sloupcově stochastická.
Pak platí

$$\|A\vec{z}\|_1 = 1$$

Důkaz.

Označme $\vec{y} = A\vec{z}$. Pak

$$\|\vec{y}\|_1 = \vec{e}^{\mathrm{T}} A\vec{z} = \vec{e}^{\mathrm{T}}\vec{z} = 1,$$

protož A je sloupcově stochastická ($\vec{e}^{\mathrm{T}} A = \vec{e}^{\mathrm{T}}$).

- Kvůli rozměrům Google matice je netriviální vypočítat součin

$$\vec{y} = A\vec{z}, \qquad \text{kde } A = \alpha P + (1-\alpha)\frac{1}{n}\vec{e}\vec{e}^{\mathrm{T}}$$

  a navíc

$$P = Q + \frac{1}{n}\vec{e}\vec{d}^{\mathrm{T}},$$

  kde d má 1 na těch pozicích, které odpovídají stránkám bez outlinků.
- Takže, abychom vytvořili P, vložíme velké množství plných vektorů do Q (každý z nich má stejnou dimenzi, jako je celkový počet web. stránek.
- Následkem toho nemůžeme uchovávat celou matici P v paměti explicitně.

Podívejme se blíže na násobení $\vec{y} = A\vec{z}$:

$$\begin{aligned}\vec{y} &= \alpha\left(Q + \frac{1}{n}\vec{e}\vec{d}^{\mathrm{T}}\right)\vec{z} + \frac{(1-\alpha)}{n}\vec{e}(\vec{e}^{\mathrm{T}}\vec{z}) \\ &= \alpha Q\vec{z} + \beta\frac{1}{n}\vec{e}\end{aligned}$$

kde

$$\beta = \alpha\vec{d}^{\mathrm{T}}\vec{z} + (1-\alpha)\vec{e}^{\mathrm{T}}\vec{z}.$$

Hodnoty β ale nemusíme počítat podle této rovnice.

Namísto toho můžeme použít $\|A\vec{z}\|_1 = 1$ a (z předchozího sladu)

$$\vec{y} = \alpha Q \vec{z} + \beta \frac{1}{n} \vec{e}$$

Dostaneme

$$1 = e^{\mathrm{T}}(\alpha Q \vec{z}) + \beta \vec{e}^{\mathrm{T}} \left(\frac{1}{n} \vec{e} \right) = e^{\mathrm{T}}(\alpha Q \vec{z}) + \beta.$$

Takže $\beta = 1 - \|\alpha Q \vec{z}\|_1$.

A jako bonus:
Nepotřebujeme ten vektor $\vec{d}$.