PageRank

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

- *outlink* = webová stránka, na kterou se odkazuje i.
- O_i množina všech outlinků i
- *inlink* = webová stránka, kter se odkazuje na i.
- Počet outlinků bude označován $N_i = |O_i|$
- I_i množina všech inlinků i.

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

problém: Ale hodnocení založené čistě na počtu inlinků se dá snadno zmanipulovat: například uměle zvyšovat důležitost stránky tak, že se vytvoří velké množství stránek s outlinky na i.

řešení: Aby se od tohoto odradilo, definuje se hodnocení (*rank*) stránky tak, že pokud stránka j s vysokým rankem má outlink na stránku i, zvýší to důležitost stránky následovně:

- 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

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

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

Tato předběžná verze je rekurzivní, a proto nemůže být vypočtena přímo. K výpočtu se používá iterační metoda. Odhadne se počáteční vektor $r^{(0)}$ ranků a pak se iteruje

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

Problém: Pokud nějaká stránka nemá žádné outlinky...

- postupně nasbírá rank přes svoje inlinky.
- její rank už pak není dále distribuován.

Takže není jisté jestli metoda konverguje.

Pro lepší vhled přeformulujeme

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

na problém hledání vlastních hodnot matice reprezentující internet.

Nechť Q je čtvercová matice dimenze n, taková že

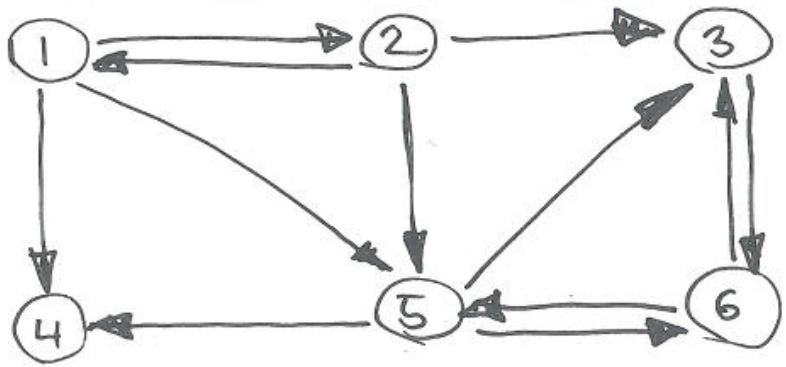
$$Q_{ij} = \begin{cases} 1/N_j & \text{pokud existuje link z } j \text{ na } i, \\ 0 & \text{jinak.} \end{cases}$$

To znamená, že

- nenulové hodnoty na řádku i odpovídají inlinkům vedoucím na stránku i;
- nenulové hodnoty ve sloupci j odpovídají outlinkům ze stránky j;
- tyto hodnoty jsou rovny $1/N_j$ a součet všech hodnot ve sloupci je 1.

Example

Následující graf reprezentuje množinu webových stránek s inlinky a outlinky



Odpovídající matice je:

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

Vzorec

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

je pak vlastně skalární součin řádku i matice Q a vektoru r.

Tuto rovnici můžeme zapsat ve tvaru

$$\lambda r = Qr, \quad \lambda = 1.$$

To znamená, že r je vlastní vektor přís. k vlastní hodnotě $\lambda = 1$.

Teď jde vidět, že iteraci

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

můžeme zapsat jako

$$r^{(k+1)} = Qr^{(k)}, \quad k = 1, 2, \ldots,$$

což je *mocninná metoda* pro výpočet vlastního vektoru.

Zatím ale nevíme, jestli je PageRank dobře definovaný.
Protože dosud nevíme, jestli existuje vlastní hodnota $\lambda = 1$.

Pro analýzu tohoto problému použijeme následující náhled.

PageRank budeme reprezentovat jako náhodnou procházku.

Předpokládejme, že nějaký návštěvník stránky (náhodný surfař) vybírá následnou stránku z outlinků aktuální stránky vždy se stejnou pravděpodobností.

Náhodný surfař by se nikdy neměl zastavit. Jinými slovy, náhodná procházka by nikdy neměla obsahovat stránku bez outlinků (taková stránka odpovídá nulovému sloupci v matici Q).

Proto je model modifikován tak, že nulové sloupce jsou nahrazeny sloupci s konstantními hodnotami ve všech pozicích. To znamená, že je stejná pravděpodobnost přechodu do jakékoli stránky na Internetu.

Definujeme vektory

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

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

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

Modifikovaná matice je pak dána takto:

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

Takováto matice P je pak **sloupcově stochastická matice**.
To znamená, že má nezáporné prvky, a prvky v každém sloupci dávají součet 1.

Toto tvrzení může být také vyjádřeno následovně:

Theorem

Sloupcově stochastická matice P splňuje

$$e^T P = e^T,$$

kde e je definováno

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

Example

Matice z předchozího příkladu

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

bude modifikovaná 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

$$\lambda r = Qr, \quad \lambda = 1.$$

bychom chtěli definovat PageRank jako unikátní vlastní vektor matice P příslušný k vlastní hodnotě $\lambda = 1$.

$$\lambda r = Pr, \quad \lambda = 1.$$

Prvek r_i na pozici i je pravděpodobnost, že náhodný surfař skončí po velkém množství kroků právě na stránce i.
(stacionární rozložení pravděpodobnosti pro Markovův řetěz)

Existence unikátního vlastního vektoru s vlastní hodnotou $\lambda = 1$ ale stále není zaručena.
K tomu potřebujeme zajistit, aby přechodová matice byla **ireducibilní**.

Definition

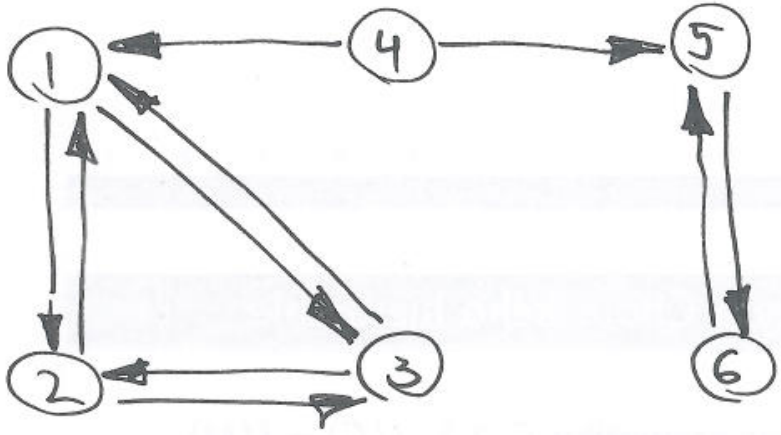
Čtvercová matice se nazývá *reducibilní*, pokud existuje permutační matice P, t.ž.

$$PAP^T = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}, \tag{1}$$

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

Example

Příklad grafu, který odpovídá reducibilní matici:



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

Graf odpovídající ireducibilní matici je *silně souvislý*: pro každé dva uzly n_i, n_j existuje cesta z n_i do n_j.

Unikátnost největší vlastní hodnoty ireducibilní, pozitivní matice je zaručena *Perron-Frobeniovou větou*;

tuto větu zde uvedeme v upravené podobě pro speciální případ, který zde uvažujeme.

značení

- $A > 0$ – že A je striktně positivní ($A_{ij} > 0$ pro všechna i, j).
- λ_1 – Dominantní vlastní hodnota = největší z vlastních hodnot

Theorem

Nechť A je ireducibilní sloupcově stochastická matice.

- $\lambda_1 = 1$;
- *existuje unikátní odp. vl. vektor r splňující $r > 0$ a $\|r\|_1 = 1$;*
- *toto je jediný vlastní vektor, který je nezáporný;*
- *pokud $A > 0$, pak dále platí $|\lambda_i| < 1, i = 2, 3, \ldots, n$.*

Vzhledem k velikosti Internetu můžeme s jistotou považovat matici P za reducibilní. To ale znamená, že Pagerankový vlastní vektor není dobře definovaný.

Abychom zajistili ireducibilitu, přidáme link z každé stránky na všechny ostatní. Z pohledu matic to můžeme udělat tak, že vezmeme konvexní kombinaci P a rank-1 matice:

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

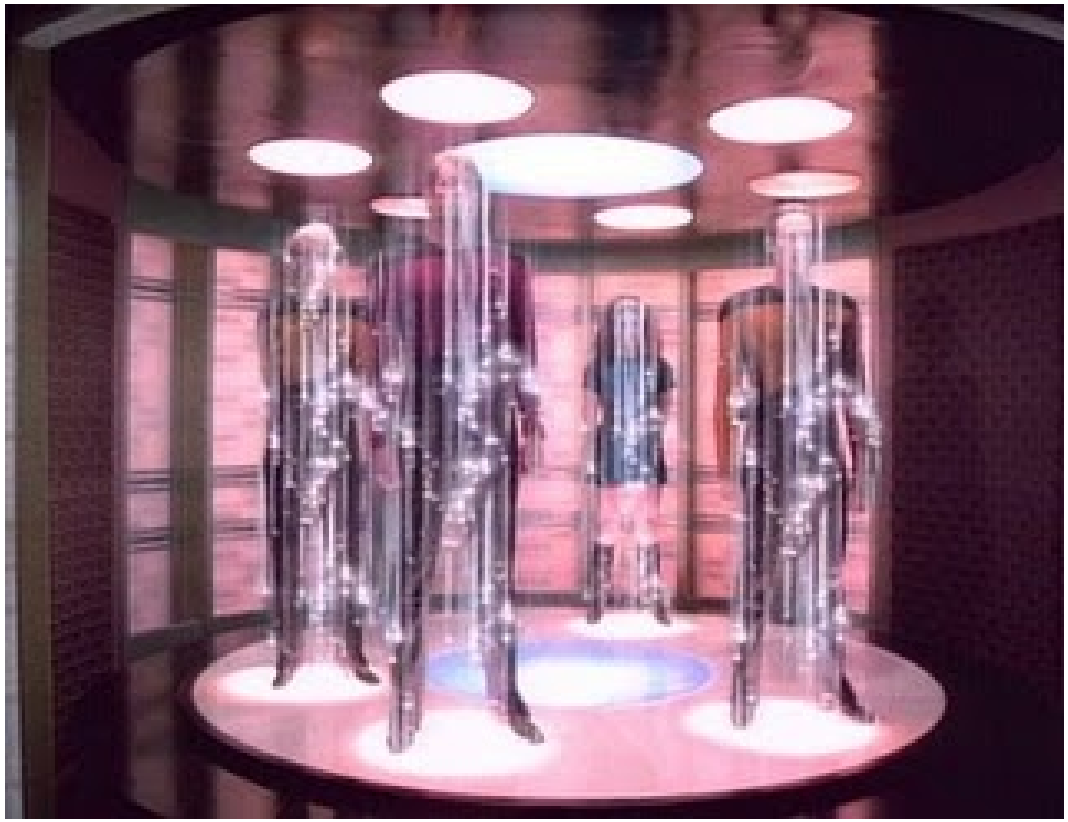
pro nějaké $\alpha \in \langle 0, 1 \rangle$.

Je zjevné, že matice A je sloupcově stochastická:

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

Interpretace 1-rank matice z pohledu náhodné procházky je, že při každém kroku je možné, že se náhodný surfař skočí na náhodně zvolenou stránku s pravděpodobnostní $1 - \alpha$.

Toto bývá někdy označováno jako *teleportace*.



Teď ukážeme, že Pagerank matice A je dobře definovaný.

Theorem

Sloupcově stochastická matice A definovaná v

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

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

Kvůli konvergenci numerického algoritmu pro výpočet vlastního vektoru je nutné prozkoumat, jak se mění vlastní hodnoty matice P při této modifikaci (na A).

Theorem

Uvažujme, že vlastní hodnoty matice P jsou $\{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$. Pak vlastní hodnoty matice $A = \alpha P + (1-\alpha)ee^T$ jsou $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_n\}$.

Důkaz

Definujme $\hat{e}$ jako normalizovaný vektor e na Euklidovsk. velikost 1, a 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 $\hat{e}^T P = \hat{e}^T$, dostáváme

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

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

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

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

A tedy

$$\begin{aligned} U^T A U &= U^T(\alpha P + (1-\alpha) v e^T) U \\ &= \alpha \begin{pmatrix} 1 & 0 \\ w & T \end{pmatrix} + (1-\alpha) \begin{pmatrix} n^{-\frac{1}{2}} \\ U_1^T v \end{pmatrix} \begin{pmatrix} n^{\frac{1}{2}} & 0 \end{pmatrix} \\ &= \alpha \begin{pmatrix} 1 & 0 \\ w & T \end{pmatrix} + (1-\alpha) \begin{pmatrix} 1 & 0 \\ n^{\frac{1}{2}} U_1^T v & 0 \end{pmatrix} =: \begin{pmatrix} 1 & 0 \\ w_1 & \alpha T \end{pmatrix}. \end{aligned}$$

Tvrzení věty pak z tohoto přímo vyplývá.

Ten Theorem říká, že i když P má více vlastních hodnot rovných 1 (což je případ Google matice), druhá největší vlastní hodnota A je vždy rovna α.

Example

Vypočítáme vlastní hodnoty a vlastí vektory matice $A = \alpha P + (1-\alpha)\frac{1}{n} e e^T$ s P z příkladu a $\alpha = 0.85$.
Kód v Octave

```
LP=eig(P).';
e=ones(6,1);
A=0.85*P+0.15/6*e*e.';
[R,L]=eig(A);
```

dává následující výsledky:

LP =

$$\begin{matrix} -0.50 & 1.00 & -0.50 & 1.00 & -1.00 & 0.00 \end{matrix}$$

R =

$$\begin{matrix} 0.45 & -0.37 & -0.35 & 0.00 & 0.82 & -0.34 \\ 0.43 & -0.37 & 0.35 & 0.00 & -0.41 & -0.47 \\ 0.43 & -0.37 & 0.35 & 0.00 & -0.41 & 0.81 \\ 0.06 & 0.00 & -0.71 & -0.00 & 0.00 & 0.00 \\ 0.47 & 0.55 & 0.00 & -0.71 & -0.00 & 0.00 \\ 0.46 & 0.55 & 0.35 & 0.71 & 0.00 & 0.00 \end{matrix}$$

diag(L) =

$$\begin{matrix} 1.00 & 0.85 & 0.00 & -0.85 & -0.43 & -0.43 \end{matrix}$$

Vidíme, že první vlastní vektor je jediný nezáporný, tak jak je to stanoveno v té větě.

Místo modifikace

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

můžeme definovat

$$A = \alpha P + (1 - \alpha)ve^T,$$

kde v je nezáporný vektor s $\|v\|_1 = 1$, který může být vybrán tak, aby ovlivňoval vyhledávání některých druhů webových stránek.

Vektoru v říká *personalizační vektor* – může být použit pro zamezení manipulace takzvanými link farmami.

Chceme řešit problém vlastní hodnoty

$$Ar = r,$$

kde r je normalizovaný $\|r\|_1 = 1$.
V této části budeme označovat hledaný vlastní vektor t_1.
Jediná schůdná metoda je *mocninná metoda*.

Předpokládejme, že je dána iniciální aproximace $r^{(0)}$.
Mocninná metoda je dána následujícím algoritmem:

for $k = 1, 2 \ldots$ until convergence **do**
$q^{(k)} \leftarrow Ar^{(k-1)}$
$r^{(k)} = q^{(k)} / \left\|q^{(k)}\right\|_1$
end for

Mocninná metoda je dána následujícím algoritmem:

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

Normalizace (tak aby $\|r\|_1 = 1$) se dělá, aby se zabránilo situaci, kdy je vektor příliš velký nebo příliš malý, nedá se pak reprezentovat v plovoucí řádové čárce.

Později uvidíme, že pro výpočet Pageranku to není potřeba. Také nepočítáme aproximaci odpovídající vlastní hodnoty, protože víme, že tato vlastní hodnota je rovna 1.

Konvergence metody záleží na rozložení vlastních hodnot.
Pro zjednodušení budeme předpokládat, že A je diagonalizovatelná;
t.j. $\exists$ regulární matice T z vlastních vektorů, t.ž.

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

Vlastí hodnoty λ_i jsou uspořádané $1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_n|$.
Rozložíme iniciální aproximaci $r^{(0)}$ podle vlastních vektorů

$$r^{(0)} = c_1 t_1 + c_2 t_2 + \cdots + c_n t_n,$$

kde c_1 se předpokládá, že $c_1 \neq 0$ a $r = t_1$ je hledaný vlastní vektor.
Pak máme

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

Je zjevné, že $\sum_{j=2}^{n} c_j \lambda_j^k t_j$ jde k nule
(protože pro $j = 2, 3, \ldots$ máme $|\lambda_j| < 1$)
a metoda konverguje k vlastnímu vektoru $r = t_1$.

Rychlost konvergence je určena $|\lambda_2|$. Pokud je tato hodnota blízká 1, je iterace velmi pomalá. U Google matice je $\lambda_2 = \alpha$.

Podmínka pro zastavení iterace se dá formulovat přes zbytkový vektor pro výpočet vlastních hodnot:

Nechť $\hat{\lambda}$ je vypočtená aproximace vlastní hodnoty a $\hat{r}$ je odpovídající aproximace vlastního vektoru. Dá se ukázat, že optimální chybová matice E pro kterou platí

$$(A + E)\hat{r} = \hat{\lambda}\hat{r},$$

splňuje

$$\|E\|_2 = \|s\|_2,$$

kde $s = A\hat{r} - \hat{\lambda}\hat{r}$.

Dá se ukázat, že optimální chybová matice E pro kterou platí

$$(A + E)\hat{r} = \hat{\lambda}\hat{r},$$

splňuje

$$\|E\|_2 = \|s\|_2,$$

kde $s = A\hat{r} - \hat{\lambda}\hat{r}$.

To znamená, že pokud je hodnota $\|s\|_2$ malá, pak vypočtená aproximace vlastního vektoru $\hat{r}$ je vlastní vektor matice $A + E$, která je blízká matici A.

Protože se při výpočtu Pageranku zabýváme pozitivními maticemi, jejichž sloupce dávají v součtu 1, je přirozené používat 1-normu.

To můžeme udělat, protože normy jsou ekvivalentní
. . . t.j. pro $\|\cdot\|_\alpha$,$\|\cdot\|_\beta$ existují konstanty m, M t.ž.

$$m\|x\|_\alpha \leq \|x\|_\beta \leq M\|x\|_\alpha$$

V obvyklém použití mocninné metody, vektor je normalizován, aby se zabránilo přetečení nebo podtečení. Teď ukážeme, že to není potřeba pro sloupcově stochastickou matici.

Theorem

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

$$\|Az\|_1 = 1$$

Důkaz.

Označme $y = Az$, pak

$$\|y\|_1 = e^T y = e^T Az = e^T z = 1,$$

protože A je sloupcově stochastická ($e^T A = e^T$). □

Kvůli rozměrům Google matice, je netriviální vypočítat součin

$$y = Az, \quad \text{kde } A = \alpha P + (1-\alpha)\frac{1}{n}ee^T.$$

Připomeňme, že matice P byla zkonstruována z matice skutečných odkazů Q jako

$$P = Q + \frac{1}{n}ed^T,$$

kde řádkový vektor d má 1 na všech těch pozicích, které odpovídají stránkám bez outlinků.

Takže abychom vytvořili P, vložíme velké množství plných vektorů do Q, a každý z těch vektorů má stejnou dimenzi, jako je celkový počet webových stránek.

Následkem toho nemůžeme uchovávat celou matici P v paměti explicitně.

Podívejme se blíže na násobení $y = Az$:

$$y = \alpha \left(Q + \frac{1}{n} e d^T \right) z + \frac{(1-\alpha)}{n} e(e^T z) = \alpha Q z + \beta \frac{1}{n} e, \qquad (2)$$

kde

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

Hodnoty β nemusíme počítat pomocí této rovnice. Namísto toho můžeme použít $\|Az\|_1 = 1$ v kombinaci s (2):

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

Takže $\beta = 1 - \|\alpha Q z\|_1$. Jako bonus pak dostáváme to, že vůbec nevyužíváme vektor d, t.j. nepotřebujeme vědět, které stránky nemají outlinky.