# Ранкиращи алгоритми за уеб майнинг

## Google PageRank, HITS, SALSA

## Google PageRank, HITS, SALSA

- Това са алгоритми, които използват матричния апарат и се основават на намиране на собствения вектор, съответстващ на най-голямата собствена стойност на матрица (*доминантен собствен вектор*, *доминантна собствена стойност*). Затова се наричат *спектрални* методи (спектър на квадратна матрица е множеството от собствените ѝ стойности).

- Те са част от методите за *анализ на връзките* (*link analysis*) на ориентирания Интернет граф, в който уеб страниците представляват върхове, а хипервръзките между страниците - ориентирани ребра. Пресмятанията се извършват чрез матрици, получени от матрицата на съседствата на този ориентиран граф.

- *A Survey of Eigenvector Methods for Web Information Retrieval*, Amy N. Langville, Carl D. Meyer.

## Кратка история на интернет търсачките преди Google

- В средата на 90-те години, с бързото нарастване на броя на уеб страниците, се налага разработването на ефективни алгоритми за подреждане (ранкиране) на резултатите от търсеното според тяхната релевантност и други критерии (популярност, престижност).

- Най-ранните идеи за подреждане на резултатите използват като критерий броя на повторенията на търсената дума в съдържанието на страниците. Твърде лесен за спамене метод.

- Altavista, Lycos били сред първите, които започнали да изучават структурата на Интернет графа и да я използват за подреждане на резултатите - ранкиране според броя на хипервръзките, сочещи към съответната страница (брой входящи ребра, in-degree на съответния връх), използван през 70-те години в анализа на цитиранията на публикации. Отново метод, който лесно се поддава на манипулиране.

- Кратка история на уеб търсачките:
  http://www.searchenginehistory.com/

  https://www.wordstream.com/articles/internet-search-engines-history

- През 1998 г. на 7-мата Международна Интернет конференция в Бризбейн, създателите на Google, Лорънс Пейдж и Сергей Брин представят своята идея за определянето на **значимост (важност, престиж, популярност, рейтинг)** на уеб страниците - *The PageRank Citation Ranking: Bringing Order to the Web*.

- В "сърцето" на техния алгоритъм, наречен **PageRank,** са заложени три ключови идеи за определяне на престижа на дадена уеб страница (първите две от тях взети на заем от теорията на анализа на социални мрежи - Филип Бонасич и въведената от него през 1972 г. мярка за централност, базирана на собствени вектори, *eigenvector centrality measure*).

## Трите идеи на основния компонент на PageRank

Дадена уеб страница има по-голям престиж, ако:

- към нея сочат много страници;
- към нея сочат страници с висок престиж;
- страниците, които сочат към тази страница, не сочат към други страници или сочат към малък брой други страници.

По настоящем е известно, че Google използва над 200 критерия (сигнали) за подреждане на резултатите от търсенето, като трите с най-висока тежест са: съдържанието на страницата, Page Rank-а на страницата и от 2016 г. mobile-friendly.

PageRank използва матрицата на съседствата на Интернет графа, която преобразува до преходна матрица на регулярна верига на Марков. Стационарното разпределение на тази верига се интерпретира като вектора, съдържащ ранковете на уеб страниците.

Тази основна идея за конструиране на PageRank алгоритъма не била съвсем нова. Още през 1976 г. Г. Пински и Ф. Нарин в публикацията си *Citation influence for journal aggregates of scientific publications* предлагат подобен метод за определяне на престижа на всяко научно списание в зависимост от броя на цитиранията, които това списание получава от останалите списания.

Те разглеждат претеглен ориентиран граф, чиито върхове са списанията, а наличието на ориентирано ребро от върха $i$ към върха $j$ с тегло $w_{ij}$ означава, че списанието $S_i$ цитира списанието $S_j$ $w_{ij}$ пъти. Матрицата на съседствата на този граф $A = (a_{ij})$, където

$$a_{ij} = \begin{cases} w_{ij}, \text{ ако списанието } i \text{ цитира } j \ w_{ij} \text{ пъти} \\ 0, \text{ ако } i \text{ не цитира } j \end{cases},$$

те преобразуват до преходна матрица $P = (p_{ij})$ на марковска верига, разпределяйки пропорционално престижа на всяко списание сред тези, които то цитира.

Тогава

$$p_{ij} = \frac{c_{ij}}{c_i},$$

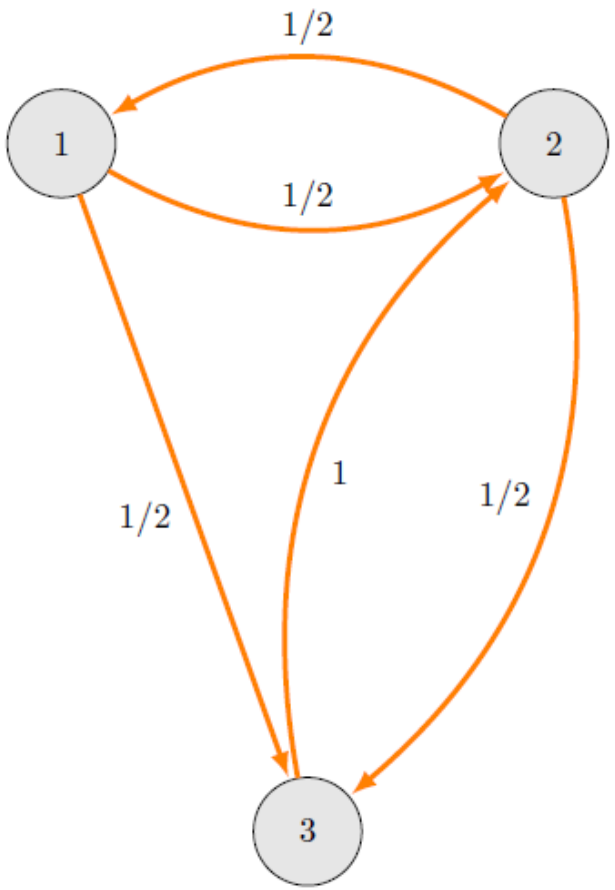
където $c_i$ е общият брой на цитиранията на списания в списанието $S_i$, а $c_{ij}$ е броят на цитиранията на списанието $S_j$ от списанието $S_i$.

Тогава, в случай, че получената марковска верига е регулярна, то $\lambda = 1$ ще бъде еднократен (прост) корен на характеристичния ѝ полином и на нея ще съответства единствен вероятностен вектор $w$ (векторът на Перон), който ще бъде стационарното и граничното разпределение на тази верига, т.е.

$$w = wP$$

Неговата $i$-та координата ще бъде равна на престижа на списанието $S_i$.

Да разгледаме следния пример с три списания. Списанието $S_1$ цитира $S_2$ и $S_3$ по един път. Списанието $S_2$ цитира списанията $S_1$ и $S_3$ по един път и $S_3$ цитира $S_2$ един път.



В този пример имаме следната преходна матрица

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

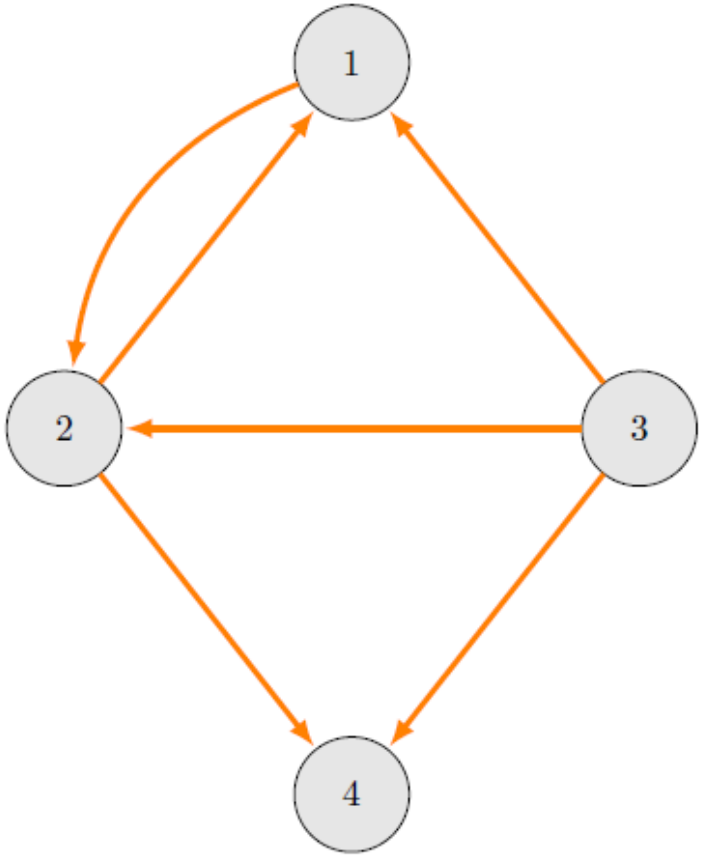
Собствените стойности на $P$ са $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = -0,5$.

За вероятностния вектор $w$, съответстващ на $\lambda_1 = 1$, получаваме

$$w = (0,22,\ 0,44,\ 0,34).$$

Най-висок престиж има списанието $S_2$, следвано от $S_3$ и $S_1$.

**Пример 1.** Нека разгледаме основната компонента на PageRank (популярността на уеб страница) чрез пример на ориентиран интернет граф от четири уеб страници и хипервръзките между тях.

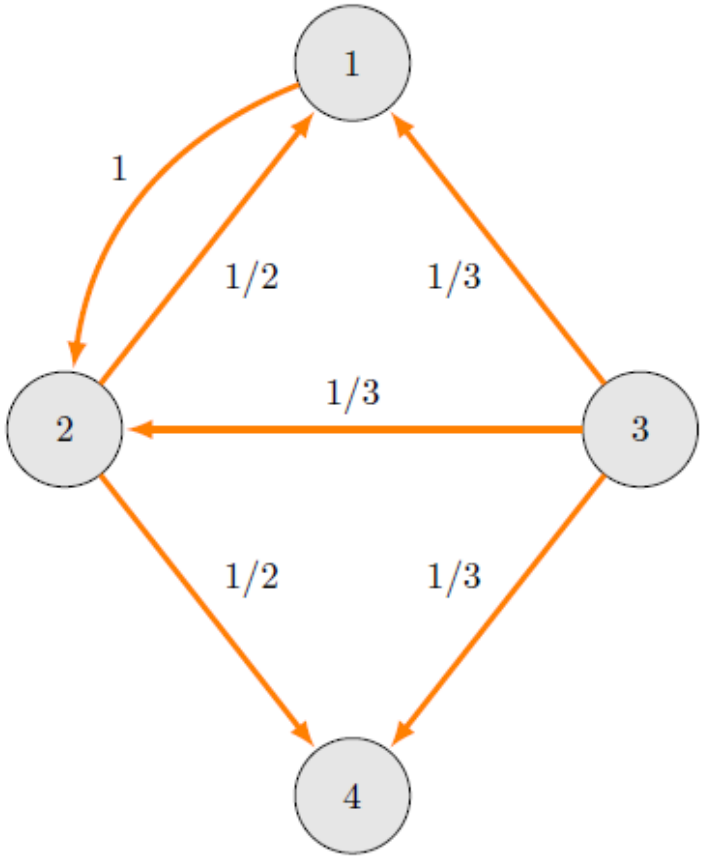



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Дадена е и матрицата на съседствата (вдясно), но ние няма да работим с нея, а ще я модифицираме до преходна матрица на регулярна марковска верига, моделирайки поведението на потребител, сърфиращ в Интернет.

Предполагаме, че когато потребител се намира в дадена страница, може да последва всеки от изходящите от страницата линкове с еднаква вероятност. Нека $n_i$ е броят на изходящите ориентирани ребра от върха $i$. Съставяме матрицата $A = (a_{ij})$, за която

$$a_{ij} = \begin{cases} \frac{1}{n_i}, & \text{ако съществува ориентирано ребро } \ i \longrightarrow j \\ 0, & \text{ако не съществува ориентирано ребро } \ i \longrightarrow j \end{cases}$$




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

Матрица $A$ не е вероятностна (стохастична), тъй като сумата от елементите в последния ѝ ред не е 1. Последният ред на $A$ е съставен само от нули, което съответства на върха (уеб страницата) 4 без изходящи линкове. Такава страница се нарича **висящ възел** (dangling node) и при предаването на ранга (престижа) между страниците в мрежата, престижът ще "изтича" от системата през такива страници.

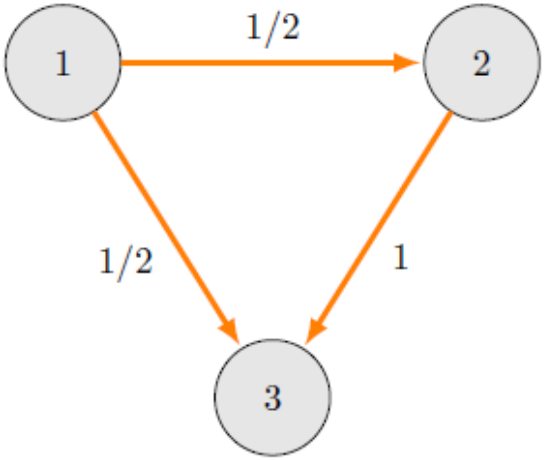
Ако използваме матрицата $A$ в този вид като преходна матрица между състоянията с произволен начален вектор, напр. вектор, в който всички страници имат равен престиж $v_0 = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$, ще забележим, че при последователно преминаване между състоянията (последователни итерации)

$$v_{k+1} = v_k A$$

стойностите на елементите на векторите $v_k$ постепенно ще намаляват, клонейки към 0.

Ето един по-драстичен пример за илюстрация на проблема с висящите възли.

**Пример 2.**




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

Нека инициираме процеса с начален вектор $v_0 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$.

$$v_1 = v_0 P = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{6} & \frac{1}{2} \end{pmatrix}$$

$$v_2 = v_1 P = \begin{pmatrix} 0 & \frac{1}{6} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{1}{6} \end{pmatrix}$$

$$v_3 = v_2 P = \begin{pmatrix} 0 & 0 & \frac{1}{6} \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}.$$

Целият начален престиж (рейтинг) на страниците "изтече" от системата през висящия възел.

Затова се налага да бъдат преобразувани нулевите редове в матрицата $A$. Пейдж и Брин използват следния подход - когато потребителят достигне до страница без хипервръзки, нека считаме, че с еднаква вероятност потребителят може да "прескочи" във всяка страница в мрежата.

По този начин нулевите редове в матрицата $A$ се заменят с редове

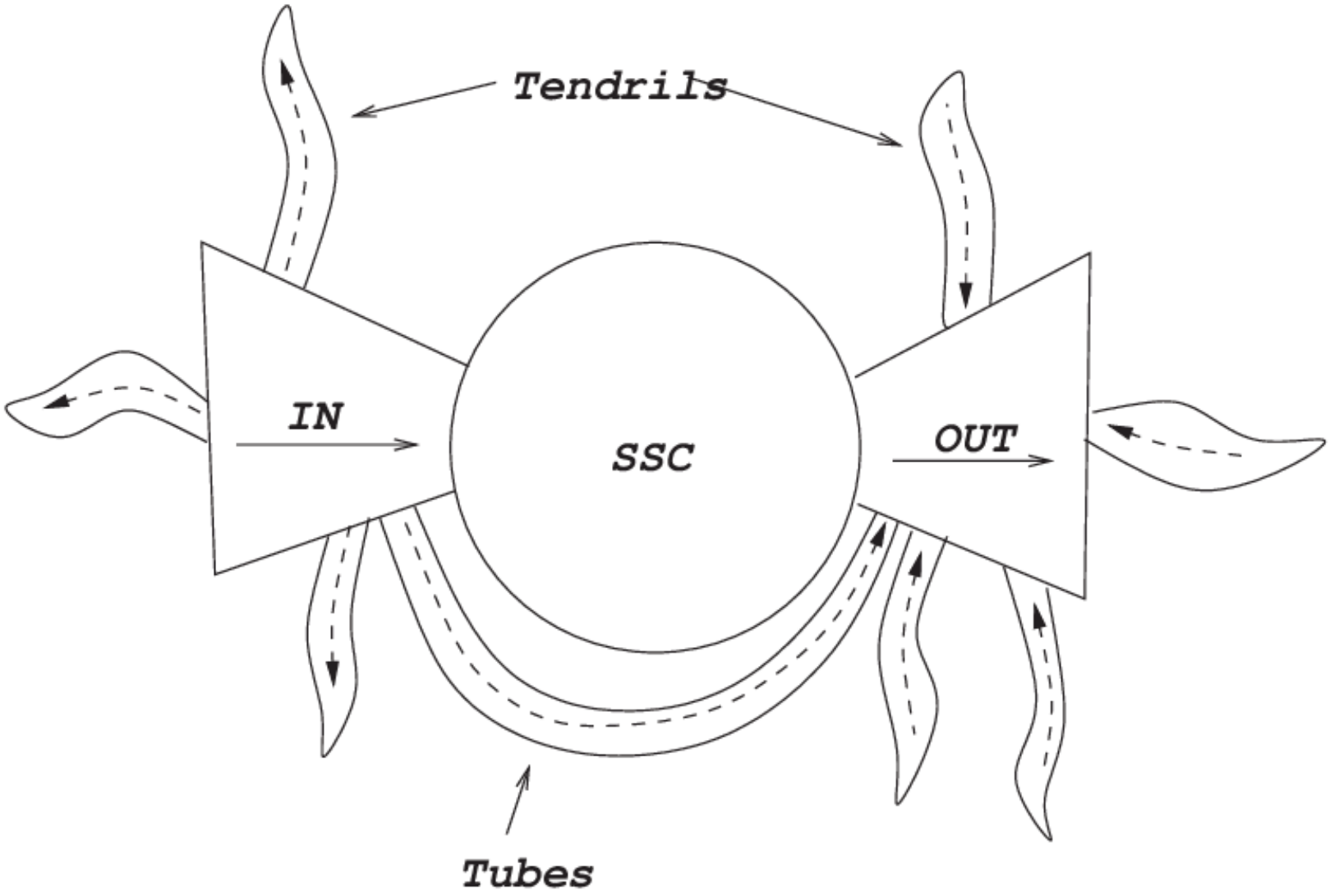
$$\left(\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N}\right),$$

където $N$ е общият брой на страниците в мрежата. За матрицата $A$ от нашия **Пример 1** това означава

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}.$$

Сега $A$ е матрица на верига на Марков, но не е сигурно дали е регулярна или дори неразложима, тъй като ориентираният граф на Интернет не е силно свързан. Остават още два проблема, с които трябва да се справим - страници без входящи линкове (sources), както и затворени цикли от страници (spider traps).

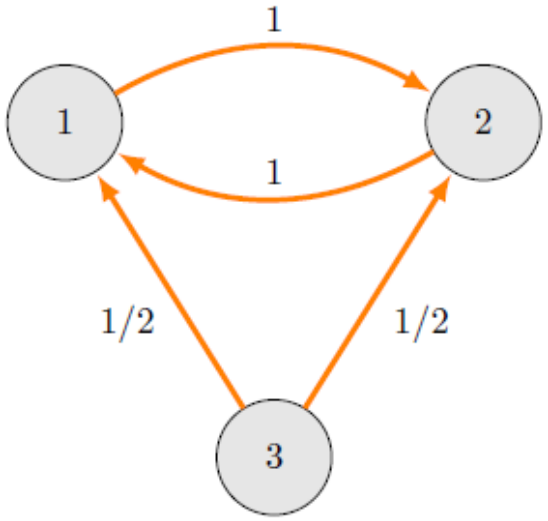
Топология на структурата на Интернет, известна като структура с форма на папионка (Broder et al., 2000). SSC - силно свързана компонента. Структурата продължава да бъде валидна.



Страниците без входящи линкове в крайна сметка ще изгубят първоначалния си престиж, защото ще го раздадат пропорционално на всички страници, към които те сочат, без да получат нищо от другите страници. Има такива страници в Интернет и те биха могли да бъдат авторитетни хъбове (да сочат към значими страници), което означава, че в подреждането на резултатите от търсенето е желателно те да бъдат сред първите предложени страници по съответната тема.

Затворени цикли са компоненти от графа, в които потребителят може да влезе, следвайки входящ линк, но от които не може да излезе, тъй като от тях няма изходящи линкове към други страници, освен страниците в тази група. Такива подграфи в крайна сметка ще акумулират престижа на всички Интернет страници и страниците в групата ще си го предават само помежду си.

**Пример 3.** В този пример ще покажем какво се случва с възлите без входящи линкове и затворените цикли.




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

Нека отново започнем с начален вектор $v_0 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$.

$$v_1 = v_0 P = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & 0 \end{pmatrix}$$

$$v_2 = v_1 P = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{6} & \frac{1}{2} & 0 \end{pmatrix}$$

$$v_3 = v_2 P = \begin{pmatrix} \frac{1}{6} & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & 0 \end{pmatrix} = v_1$$

$$v_4 = v_3 P = v_2, \quad v_5 = v_4 P = v_3, ...$$

Виждаме, че $v_{k+2} P = v_k P$, $k \geq 1$.

Третата страница без входящи линкове бързо губи целия си престиж, като той се акумулира от цикъла, в който участват първата и втората страница. Те задържат целия престиж на системата и го предават една на друга.

Пейдж и Брин решават и двата проблема едновременно с добавянето на още една вероятностна матрица и съставяне на линейна комбинация от втората матрица и $A$, предполагайки, че във всеки един момент Интернет потребителят може да направи избор между две действия - да последва изходящ линк от страницата, в който се намира, или да отвори произволна нова страница (teleport).

За тази цел съставяме квадратна матрица от ред $N \times N$ ($N$ е общият брой на страниците в Интернет), всеки елемент на която е равен на $\frac{1}{N}$ - вероятността да бъде избрана произволна уеб страница, предполагайки, че тези вероятности са еднаква за всяка страница и за всеки потребител. В този случай матрицата се нарича неутрална. Но тази матрица би могла и да се персонализира (personalized PageRank) в зависимост от предпочитанията на всеки отделен потребител относно различни области на интереси.

За малкия Интернет граф от 4 страници неутралната матрица има вида

$$B = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}.$$

Нека отбележим, че $B$, като $A$, е също вероятностна матрица.

След това съставяме линейна комбинация на матриците $A$ и $B$

$$G = \alpha A + (1 - \alpha)B,$$

където $\alpha \in (0; 1)$ е вероятността потребителят да последва изходящ линк от страницата, в който се намира, а $1-\alpha$ е вероятността да отвори произволна нова страница. Тъй като сумата от коефициентите в линейна комбинация на вероятностните матрици $A$ и $B$ е равна на 1, то матрицата $G$ също е вероятностна.

$G$ са нарича *матрица на Google*.

Нещо повече - $G$ е преходна матрица на регулярна марковска верига. Тогава, прилагайки теорията на веригите на Марков, $G$ има единствен стационарен вероятностен вектор $\pi = \pi G$ (собствен вектор, съответстващ на най-голямата собствена стойност на $G$, $\lambda = 1$). Този вектор, както знаем, съвпада с граничното разпределение на веригата, т.е. с вектора $\widehat{v} = \lim_{k \to \infty} v_0 G^k$, където $v_0$ е произволен начален вероятностен вектор (произволно начално състояние на системата).

Тогава, разглеждайки сърфирането из Интернет като процес на Марков с преходна матрица $G$, елементите на стационарния вектор $\pi$ могат да се интерпретират като дългосрочната вероятност всяка страница да бъде посетена от потребителите.

Именно координатите на вектора $\pi$ са PageRank-овете на страниците в мрежата.

**Каква е стойността на реалния параметър $\alpha$?**

В оригиналната публикация на Пейдж и Брин $\alpha$ е избрана да бъде $\alpha = 0.85$ и този избор не е случаен.

Стойност на $\alpha$, близка до 1, означава, че се дава по-голяма тежест на структурата от хипервръзки на Интернет (какъвто е първоначалният замисъл на алгоритъма), т.е. на матрицата $A$. Стойност на $\alpha$, близка до 0, означава, че се дава по-голяма тежест на "прескачането" между страници (матрицата $B$).

Но стойност на $\alpha$, твърде близка до 1, създава изчислителни затруднения. Методът, използван за пресмятане на PageRank, е матричният итерационен метод (Matrix power method), съгласно който се пресмятат последователно векторите $v_k$

$$v_{k+1} = v_k G, \qquad k \geq 0,$$

започвайки от начално приближение $v_0$, до достигане на желаната точност, т.е. до $|v_{k+1} - v_k| < \epsilon$.

Ако $\{u_1, u_2, ..., u_n\}$ е база от собствени вектори на матрицата $G$, отговарящи на собствените стойности $\lambda_1$, $\lambda_2$,..., $\lambda_n$ (за удобство $\lambda_i \in \mathbb{R}$), т.е. $u_i G = \lambda_i u_i$, то за всеки вектор $v$ е възможно представянето

$$v = c_1 u_1 + c_2 u_2 + ... + c_n u_n.$$

Тогава

$$vG = c_1 u_1 G + c_2 u_2 G + ... + c_n u_n G = c_1 \lambda_1 u_1 + c_2 \lambda_2 u_2 + ... + c_n \lambda_n u_n.$$

Следователно

$$\begin{aligned} vG^k &= c_1 \lambda_1^k u_1 + c_2 \lambda_2^k u_2 + ... + c_n \lambda_n^k u_n \\ &= \lambda_1^k \left[ c_1 u_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k u_2 + ... + \left(\frac{\lambda_n}{\lambda_1}\right)^k u_n \right]. \end{aligned}$$

Нека $|\lambda_1| > |\lambda_i|$ за всяко $i = 2, 3, .., n$. В този случай се казва, че $\lambda_1$ е *доминантна собствена стойност* ($\lambda_1 \in \mathbb{R}$) и съответният ѝ собствен вектор също се нарича *доминантен*.

В такъв случай при $k \to \infty$ в последното равенство, всеки от коефициентите $\left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0$, $i = 2, 3, .., n$. Така получаваме

$$vG^k \to c_1 \lambda_1^k u_1, \quad k \to \infty.$$

Това представлява матричният итерационен метод, който се използва за пресмятане на вектора, съдържащ ранга на всички страници.
Нека си припомним, че за преходни матрици на регулярни марковски вериги, каквато е $G$, имаме $|\lambda_i| < 1 = \lambda_1$, т.е. $vG^k \to c_1 u_1$, от който след нормиране получаваме стационарното (граничното) разпределение.

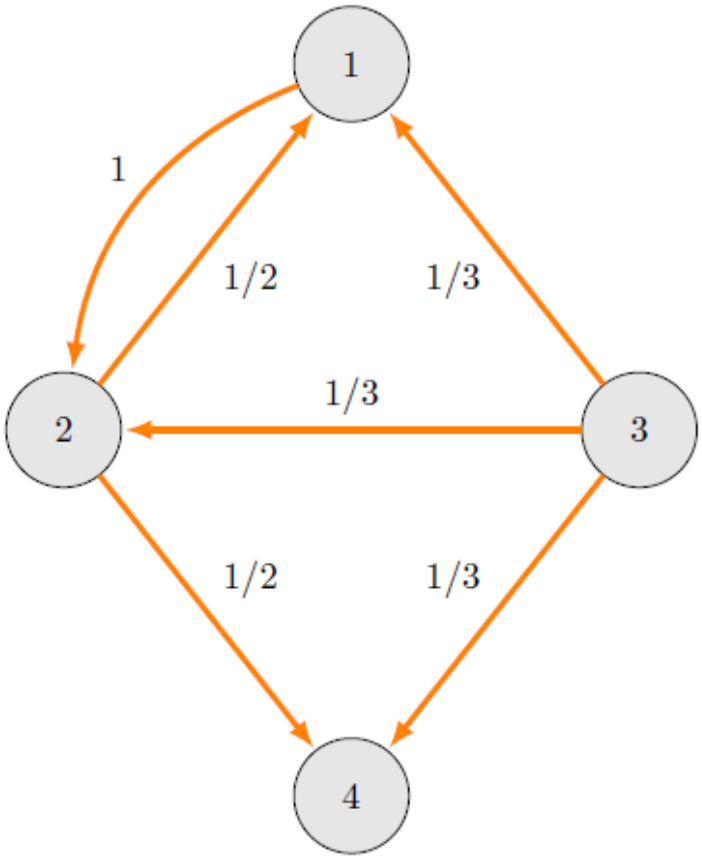
Ако с $\lambda_2$ означим собствената стойност, която по модул е най-близка до 1 (субдоминантната собствена стойност), то бързината, с която чрез този метод ще се достигне до крайния резултат (стационарното разпределение) зависи от стойността на отношението $\frac{\lambda_2}{\lambda_1}$.

В публикацията си *The Second Eigenvalue of the Google Matrix*, Taher H. Haveliwala, Sepandar D. Kamvar доказват, че за субдоминантната собствена стойност на матрицата на Google $G = \alpha A + (1 - \alpha)B$ е изпълнено $|\lambda_2| \leq \alpha$. Оттук идва необходимостта да се направи компромис между това стойността на параметъра $\alpha$ да бъде близка до 1, но не твърде близка.

В статията си Пейдж и Брин съобщават, че за изчисляване на вектора на PageRank с точност между $10^{-3}$ и $10^{-7}$ са им били необходими между 50 и 100 итерации на матричния метод.

Освен това при стойности на $\alpha$, близки до 1, малки изменения в структурата от хипервръзки в Интернет (които се случва ежедневно) могат да създават по-значими промени в подреждането на страниците по PageRank-а им.

Нека се върнем на **Пример 1** и да намерим стационарното разпределение (PageRank вектора) на Интернет графа при $\alpha = 0.85$.




$$\pi = (0.274158,\ 0.355925,\ 0.0957586,\ 0.274158)$$

От получения резултат се вижда, че най-висок престиж има страница № 2, след нея на второ място се нареждат № 1 и № 4, а № 3 е на последно място.

## Предимства на PageRank

- По-ефективно се справя със спама в сравнение с метода, ранкиращ страниците само въз основа на броя на линковете, сочещи към тях (in-degree ranking).

- Методът на Google е независим от търсенето (query independent) - рейтингът на страниците е предварително пресметнат (обновява се всеки месец) и след получаване на резултатите от всяка потребителска заявка за търсене, релевантните по съдържание страници се подреждат съгласно рейтинга си. С това се печели време при предоставяне на резултатите от търсенията на потребителите.

## Недостатъци на PageRank

- Трудности при разграничаване между страници, които са престижни по темата на търсенето и престижни по принцип. PageRank-ът на всяка страница е еднозначно определен, независимо от тематичното ѝ съдържание.

- До някаква степен фаворизира по-стари страници, тъй като към тях би могло да има повече хипервръзки, докато по-новите страници, които не са имали достатъчно време, за да бъдат цитирани от други страници, биха могли да имат по-нисък рейтинг. Едно решение на този проблем - TS-PageRank (Time Sensitive PageRank). Представлява модификация на Page Rank, в която вместо $\alpha = \text{const.}$ се използва функция на времето $f(t_i)$ $(0 \leq f(t_i) \leq 1)$, където $t_i$ е разликата между времето на пресмятане на PageRank и времето, когато страницата $i$ е създадена или последно обновена. Например $f(t_i) = (0.5)^{\frac{t_i}{3}}$.

Въпреки предимствата си и подобренията на Google (Panda от 2011 и Penguin от 2012 г.), PageRank не е напълно защитен от спам. Някои от стратегиите за манипулиране на алгоритъма са: "ферми за линкове" и сибил атаки (sybil attacks).

При т. нар. сибил атака собственикът на страница, чийто PageRank иска да повиши изкуствено, премахва всички изходящи линкове от страницата си, създава нови фалшиви страници (наречени сибили) и свързва всеки от сибилите само със своята основна страница с изходящ и входящ линк, т.е. всеки от сибилите сочи само към основната страница и основната страница сочи към всеки сибил. По този начин се създава затворена група от върхове в Интернет графа, която акумулира PageRank.

Повече информация можете да намерите в тази публикация: *Manipulability of PageRank under Sybil Strategies*.

В публикацията си *The Second Eigenvalue of the Google Matrix*, Taher H. Haveliwala, Sepandar D. Kamvar доказват, че за субдоминантната собствена стойност на матрицата на Google $G = \alpha A + (1 - \alpha)B$ е изпълнено $|\lambda_2| \leq \alpha$, а в случай, че $A$ има поне две неразложими затворени подмножества, то $\lambda_2 = \alpha$.

Доказано е следното твърдение. Нека $u = (x_1, x_2, ..., x_n)$ е собствен вектор на $G$, отговарящ на собствената стойност $\alpha$. Тогава $x_j = 0$, точно когато върхът $j$ не принадлежи на неразложимо затворено подмножество. По този начин собствен вектор, съответстващ на субдоминантната собствена стойност, може да се използва за откриване на изкуствено създадени връзки между страници, целящи манипулиране на PageRank-а им (link spamming).

## Reverse PageRank

- За разлика от PageRank не се интересуваме от входящите ориентирани ребра в даден възел, а от изходящите от него. Затова вместо матрицата на съседствата $A$ модифицираме нейната транспонирана матрица $A^T$.
- PageRank дава информация кои са значимите върхове. Reverse PageRank дава информация защо даден връх е значим. Колкото по-висок Reverse PageRank има даден връх, толкова повече върхове от графа могат да бъдат достигнати от него чрез ориентирани пътища.
- Може да се разглежда като модел на потребител, сърфиращ из Интернет, който вместо да следва изходящите връзки като при PageRank, следва входящите.
- Алгоритъм на Twitter за определяне на влиятелни потребители в социалната мрежа се базира на Reverse PageRank.

## HITS (Hypertext Induced Topic Search)

- Разработен от Джон Клайнбърг през 1997 г. малко след PageRank. Първоначално описан в публикацията му *Authoritative Sources in a Hyperlinked Environment*.

- Докато PageRank е статичен алгоритъм (query independent), HITS е динамичен (query dependent).

- Всяка страница получава два рейтинга - като източник на връзки към страници с релевантна на търсенето информация (хъб) и като престижна страница по темата на търсенето (авторитет).

- Престижни хъбове са страници, които сочат към престижни авторитети. Престижни авторитети са страници, към които сочат престижни хъбове.

При търсенето от Интернет графа се отделят определен брой най-релевантни по съдържание страници (200 на брой в оригиналната публикация на Клайнбърг). След това така полученият подграф $W$ се разширява чрез включване на определен брой страници, към които сочат страници от $W$ и които сочат към страници от $W$. За всяка страница от $W$ се включват още най-много $k$ на брой страници (Клайнбърг използва $k = 50$) и по този начин се получава по-голям ориентиран подграф $S$.

Изчисленията се извършват върху $S$.

Нека $L$ е матрицата на съседствата на ориентирания граф $S$. Нека с $h$ и $a$ означим съответно векторите, съдържащи рейтинга като хъб и като авторитет на всяка страница от $S$. Тогава

$$h = \alpha L a, \qquad a = \beta L^T h, \qquad \alpha, \beta \in \mathbb{R}.$$

$$(LL^T)h = \lambda h, \qquad (L^T L)a = \lambda a,$$

където $\lambda = (\alpha\beta)^{-1}$.

Векторите $h$ и $a$ са доминантните десни собствени вектори съответно на матриците $H = LL^T$ (матрица на хъбовете) и $A = L^T L$ (матрица на авторитетите), отговарящи на доминантната собствена стойност на двете матрици $H$ и $A$.

Ненулевите собствени стойности на $H = LL^T$ и $A = L^T L$ съвпадат. Матриците $H$ и $A$ са симетрични, неотрицателни и положително полудефинитни. Следователно всичките им собствени стойности са реални неотрицателни числа, откъдето следва, че съществува доминантна собствена стойност $\lambda_1$, за която $\lambda_1 > \lambda_i$ за всяка друга собствена стойност $\lambda_i$. Това условие гарантира, че итерационният метод, използван за пресмятане на векторите $h$ и $a$, е сходящ.

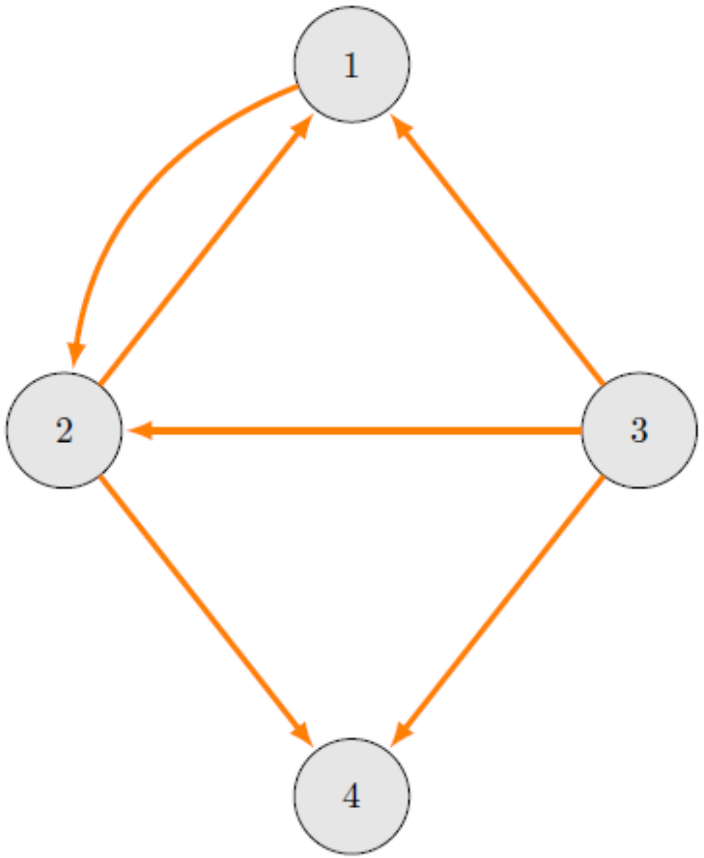
Този итерационен алгоритъм е следният. Започва се с начален вектор $h_0$. След това на всяка стъпка $k \geq 1$ се пресмята

$$a_k = L^T h_{k-1}, \qquad h_k = L a_k$$

като всеки от получените вектори $a_k$ и $h_k$ се нормират. Пресмятането се извършва до достигане на необходимата точност.

Съществената разлика с PageRank е, че доминантната собствена стойност $\lambda_1$ може да не е прост корен, а кратен корен на характеристичния полином и тогава на нея ще съответства многомерно пространство от собствени вектори (размерност 2 или по-голяма). Последното означава, че при различни начални вектори $h_0$ ще се получават различни резултати за $h$ и $a$. Дължи се на факта, че матрицата $LL^T$ (и съотв. $L^T L$) е възможно да не бъде неразложима (такива матрици се наричат *разложими*).

**Пример 1.** Нека разгледаме как работи HITS в този ориентиран граф.




$$L = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$H = LL^T = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 2 & 0 \\ 1 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A = L^T L = \begin{pmatrix} 2 & 1 & 0 & 2 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 2 \end{pmatrix}.$$

Собствените стойности на двете матрици са $\lambda_1 = 3 + \sqrt{3}$, $\lambda_2 = 3 - \sqrt{3}$, $\lambda_3 = \lambda_4 = 0$.

Собственият вектор на $H = LL^T$, отговарящ на най-голямата собствена стойност, е

$$h = (0.267949, \ \ 0.732051, \ \ 1, \ \ 0).$$

Координатите на $h$ са рейтингите на четирите страници като хъбове. Собственият вектор на $A = L^T L$, отговарящ на най-голямата собствена стойност, е

$$a = (1, \ \ 0.732051, \ \ 0, \ \ 1).$$

Координатите на $a$ са рейтингите на страниците като авторитети.

## Предимства на HITS

- Всяка страница получава два вида рейтинг - като хъб и като авторитет.
- Изчисленията се извършват върху по-малки матрици.

## Недостатъци на HITS

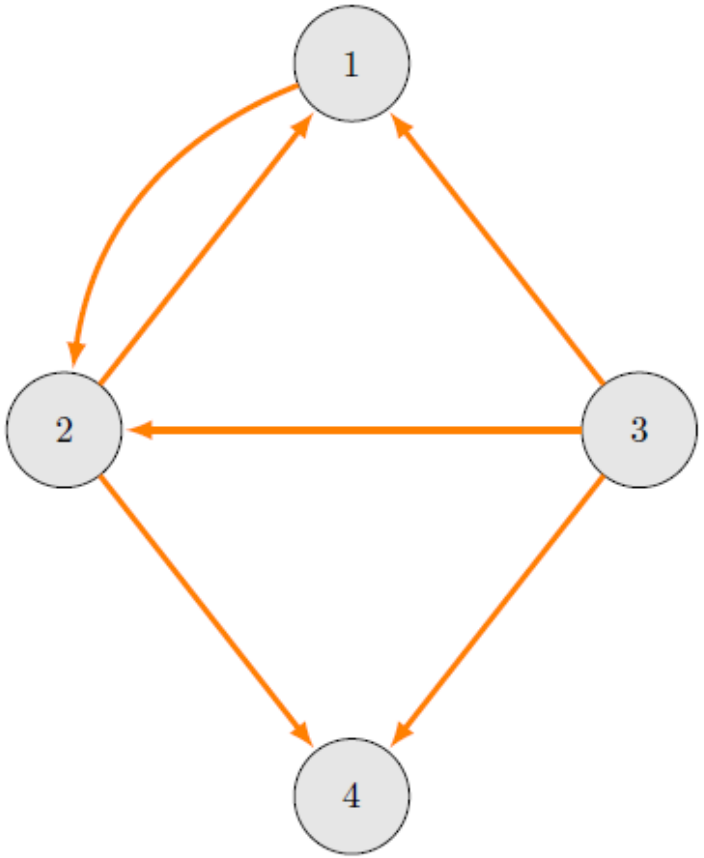
- Подграфът, с който се работи, се създава при обработване на потребителската заявка (query dependent).
- Отклоняване от темата на търсенето (Topic drift) при разширяване на първичния подграф $W$ до $S$.
- По-лесно податлив на манипулация в сравнение с PageRank. Тъй като ранговете на хъбовете и авторитетите са взаимно свързани, чрез добавяне на изходящи линкове към страницата си, потребител може да увеличи ранга ѝ като хъб и оттам и като авторитет.

## SALSA - Stochastic Approach to Link Structure Analysis

- Разработен през 2000 г. от Лемпел и Моран.
- Съчетава предимствата на HITS и PageRank. Като HITS присвоява два типа рейтинг на страниците - като хъбове и като авторитети. Подобно на PageRank използва вериги на Марков.
- Недостатъци: както при HITS всички изчисления се извършват към момента на търсенето (query dependent); матриците на марковските вериги могат да се окажат разложими (крайният резултат зависи от началното състояние).
- Разработеният през 2012-13 г. рекомендационен алгоритъм за препоръчване на потребители за следене WTF (Who To Follow) на Twitter се основава на SALSA. Хъбовете са потребители, сходни на даден потребител, а авторитетите (препоръчваните потребители за следене) са тези, които хъбовете следят.

Както при HITS по време на търсенето от Интернет графа се отделя ориентиран подграф, за който се съставя матрицата на съседствата $L$.

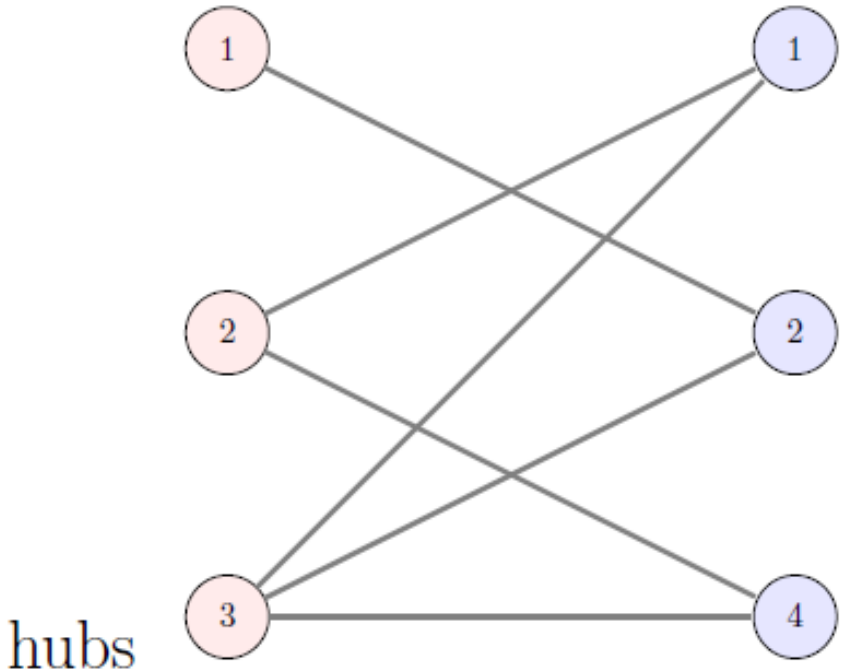
Нека приложим SALSA за графа от **Пример 1.**




$$L = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

От отделения подграф се съставя *двуделен граф*, като страниците се разделят на два типа - хъбове (източници) и авторитети.

*Двуделен граф* се нарича граф, върховете на който са разделени на две подмножества и всеки възел от дадено подмножество може да бъде свързан чрез ребро само с възли от другото подмножество.




$$L = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

В нашия пример хъбовете са $V_h = \{1, 2, 3\}$, т.е. възлите с поне едно изходящо ориентирано ребро, а авторитети са $V_a = \{1, 2, 4\}$, т.е. върховете с поне едно входящо ориентирано ребро. Всеки връх (възел) може да участва и двете подмножества.

От матрицата на съседствата $L$ се съставят две матрици: $L_r$ и $L_c$. Матрицата $L_r$ се получава, като елементите от всеки ненулев ред на $L$ се разделят на сумата на всички елементи в този ред. Аналогично, $L_c$ се получава, като елементите във всеки ненулев стълб на $L$ се разделят на сумата на всички елементи в този стълб.

$$L_r = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad L_c = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Матриците $H$ и $A$, съответно на хъбовете и авторитетите, се получават съответно от $L_r L_c^T$ и $L_c^T L_r$ след премахване на всички нулеви редове и стълбове в двете матрични произведения:

$$L_r L_c^T = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad L_c^T L_r = \begin{pmatrix} \frac{5}{12} & \frac{1}{6} & 0 & \frac{5}{12} \\ \frac{1}{6} & \frac{2}{3} & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 \\ \frac{5}{12} & \frac{1}{6} & 0 & \frac{5}{12} \end{pmatrix}.$$

След премахване на 4-тия ред и стълб в $L_r L_c^T$ и на 3-тия ред и стълб в $L_c^T L_r$ получаваме съответно

$$H = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \end{pmatrix}, \qquad A = \begin{pmatrix} \frac{5}{12} & \frac{1}{6} & \frac{5}{12} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{5}{12} & \frac{1}{6} & \frac{5}{12} \end{pmatrix}.$$

И двете матрици $H$ и $A$ са стохастични (матрици на марковски вериги). В общия случай обаче не е гарантирано, че са неразложими (ергодични) или регулярни (в нашия пример се оказват такива).

Чрез матрицата $H$ намираме ранговете като хъбове на върховете от $V_h = \{1, 2, 3\}$, а чрез матрицата $A$ ранговете като авторитети на върховете от $V_a = \{1, 2, 4\}$. И в двата случая, като при PageRank, търсим доминантния ляв собствен вектор, съответстващ на собствената стойност 1 (стационарното разпределение на марковските вериги в случай, че са ергодични).

Намираме

$$h = (0.166667,\ 0.333333,\ 0.5), \quad a = (0.333333,\ 0.333333,\ 0.333333).$$

Най-висок ранг като хъб има страница № 3, а като авторитети страниците 1, 2 и 4 имат равни рангове.