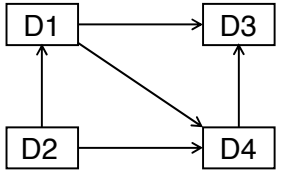


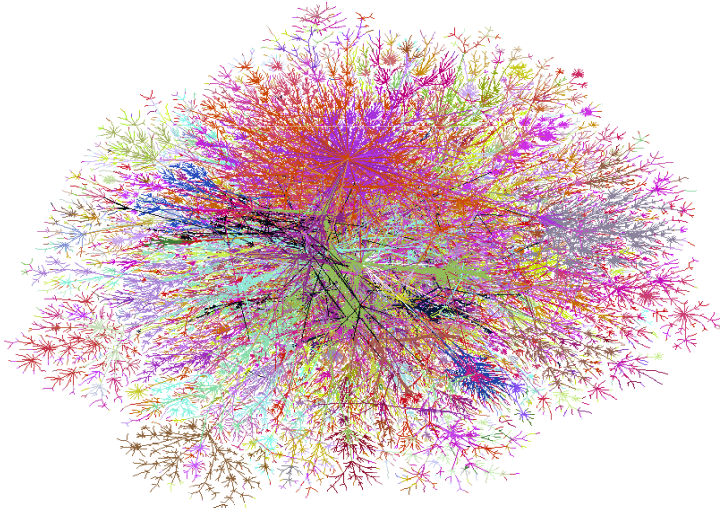
Information Retrieval and Search: Web Linkage Mining

Miłosz Kadziński
Institute of Computing Science
Poznan University of Technology, Poland
www.cs.put.poznan.pl/mkadzinski/wpi

[1] W zakresie eksploracji zasobów Internetu wyróżnia się trzy podstawowe przedmioty zainteresowania: analizę zawartości sieci (ang. web content mining), analizę jej użytkowania (ang. web usage mining) i wreszcie analizę samej struktury sieci (ang. web structure mining). Tematem wykładu jest to ostatnie zagadnienie. Skupimy się więc na analizie połączeń między strony, a w szczególności na algorytmach umożliwiających wypracowanie rankingu stron pod względem struktury sieci. Jest to przydatne nie tylko do uszeregowania stron w wyniku działania wyszukiwarki, ale także w innych dziedzinach. Podczas wykładu zostaną omówione dwa podstawowe algorytmy: PageRank oraz Hubs and authorities znany też jako HITS. Ten pierwszy algorytm należy do grupy dziesięciu najsławniejszych i najbardziej przydatnych algorytmów w dziedzinie analizy danych. W związku z tym omówimy też jego wybrane rozszerzenia. Kluczowy dla działaniach tych metod jest fakt, że ocena danej strony nie zależy od tego co sama ono o sobie twierdzi, ale raczej od jakości stron, z którymi jest powiązana.

Web Mining

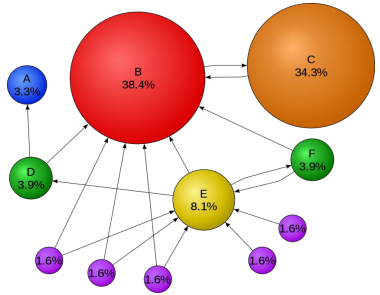
- Discovery of interesting, potentially useful knowledge contained in the ***web structure***, content and *usage*



[2] Eksploracja zasobów Internetu polega na odkrywaniu interesującej, potencjalnie użytecznej wiedzy ukrytej w zawartości sieci, jej połączeniach oraz sposobie korzystania z niej przez użytkowników. Wiedza ta jest najczęściej ujmowana w postaci dedykowanych parametrów, reguł, wzorców lub zależności. Internet i jego zasoby są ogromnym repozytorium różnorodnej wiedzy. Klasyfikacja metod stosowanych do jej ujęcia jest trudna, ale zasadniczo wyróżnia się trzy podstawowe grupy metod eksploracji danych dotyczące analizy jej zawartości, połączeń lub korzystania. Chcąc eksplorować strukturę połączeń musimy spojrzeć na sieć z dużej odległości, ignorując zarówno zawartość, jak i to w jaki sposób ze stron korzystają użytkownicy. Z takiej perspektywy widać tylko strony oraz linki między nimi.

Web Linkage Mining

Web can be seen as a directed graph



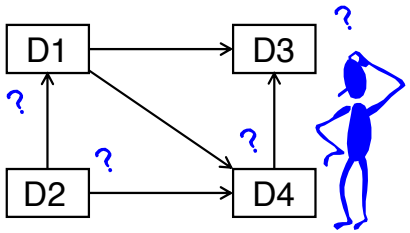
- **pages** correspond to the **nodes**
- **links** correspond to the **edges**

- links into a page are called **inlinks** and point into nodes
- links from a page are called **outlinks** and point out from nodes
- the content is not important
- thick (average degree: 8-10) graph and loose at the same time
- exponential law with respect to the ingoing and outgoing links

[3] Sieć będziemy postrzegać jako graf skierowany, w którym strony można interpretować jako wierzchołki, a linki między nimi jako łuki. Z perspektywy określonego wierzchołka wyróżniamy łuki wchodzące (ang. inlinks) oraz wychodzące (ang. outlinks). Te pierwsze pochodzą od stron, które linkują do strony reprezentowanej przez dany wierzchołek, a te drugie idą do stron, które są linkowane przez tą stronę. Graf sieci internetowej jest bardzo specyficznym zasobem. Z jednej strony, można powiedzieć, że jest on stosunkowy gęsty. Średni stopień wierzchołka wskazujący na liczbę linków wchodzących i wychodzących wynosi ok. 8-10. Z drugiej strony, jest on bardzo rzadki, jeśli wziąć pod uwagę jak niewielki jest udział tych kilku stron w całkowitej liczbie stron znajdujących się w Internecie. Tym co jeszcze wyróżnia graf Internetu obok bardzo dużego rozmiaru, jest ogromna dynamika zmian. Nowe strony i linki pojawiają się dynamiczna, a znikają te istniejące. My skupimy się jednak na analizie statycznego obrazu tego grafu w danym momencie.

Ranking Algorithms (1)

Ranking of web pages – what do we need it for?



- Order websites in the search engine results
- Adjust the crawling policy to the importance of pages
- Show your importance (Google Toolbar)
- Impact of some community on the entire web (Blogosphere)

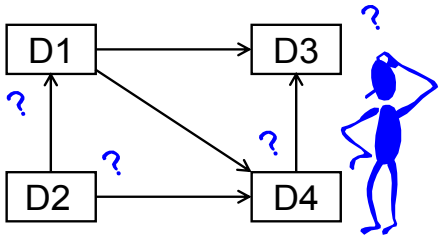
- Inspring for other domains (analysis of graphs and networks), e.g.:
  ecosystem (species essential to the continuing of its health)
  Twitter (present users with other accounts to follow)
  decision support (alternatives evaluated in terms of multiple criteria)
  bibliometrics (Eigenfactor; the importance of citations)

[4] Algorytmy eksploracji struktury połączeń sieci mają na celu wypracowanie rankingu stron. Początkowym celem badań w tym zakresie było uporządkowanie wyników wyszukiwania stron WWW. Okazało się jednak, że opracowane techniki są bardzo przydatne do dostosowania polityki crawlowania, określenia wpływu określonych środowisk lub tematyki na resztę Internetu czy do znajdowania lustrzanych serwerów web, co jest istotne w optymalizacji zapytań dla sieci rozległych. Z biegiem czasu zorientowano się też, że podobne algorytmy znajdują zastosowanie w zupełnie innych dziedzinach, na przykład w analizie relacji między gatunkami w określonym ekosystemie, użytkownikami w sieciach społecznościowych, serwisach aukcji internetowych lub systemach rekomendacyjnych czy wariantami w wielokryterialnym wspomaganiu decyzji. Szczególnie interesujący jest to przykład bibliometryki, w zakresie której podobne algorytmu były stosowane wiele lat przed powstaniem Internetu.

Ranking Algorithms (2)

Ranking of web pages

- Ranking of scientific papers (1970s):
  a paper is important if it is cited by many other important papers
- Hollywood rule:
  your importance in the show-business depends
  on the importance of stars you know



- **PageRank**
- TrustRank
- Inverse PageRank
- **HITS** (Hubs & Authorities)

[5] Na początku lat 70-tych zaproponowano metody oceny artykułów i czasopism naukowych w oparciu o cytowania. Podstawowa zasada kierująca tymi metodami mówiła, że artykuł jest istotny o ile jest cytowany przez inne artykuły i dobrze by były to artykuły istotne. Na tej bazie opierają się choćby takie miary jak impact factor lub eigenfactor. Podobna zasada funkcjonuje w Hollywood. Twoje istnienie i ważność w świecie showbiznesu zależy od ważności gwiazd, które znasz i które znają ciebie. Na podobnych zasadach opierają się algorytmy eksploracji struktury połączeń sieci. Zilustrujemy je dwoma najpopularniejszym procedurami jakimi są PageRank oraz HITS. Dodatkowo na bazie PageRank'a zbudujemy dwa inne algorytmy: TrustRank oraz Inverse PageRank, które przy niewielkiej modyfikacji oryginalnego pomysłu pozwolą na osiągnięcie innych celów i wypracowanie rankingów stron w trochę innym kontekście.

PageRank (1)

Sergey Brin, Larry Page (1998)
Stanford University, Google

"The anatomy of a large-scale hypertextual Web search engine", Computer Networks and ISDN Systems. ***30****:107–117*

”PageRank relies on the uniquely democratic nature of the Web by using its vast link structure as an indicator of an individual page’s value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important".”

[6] PageRank jest jednym z najbardziej istotnych algorytmów analizy danych w historii. Jego twórcami są Sergey Brin oraz Larry Page, a od nazwiska tego drugiego (a nie od strony internetowej - page) pochodzi nazwa metody. Brin i Page są twórcami Google, a PageRank leżał u podstaw sukcesu tej wyszukiwarki w pierwszych latach istnienia. Algorytm bazuje na “demokratycznej" naturze sieci, w której struktura linków jest wykorzystywana do wypracowania ważności poszczególnych stron. W tym kontekście Google interpretuje link od strony A do B jako głos oddany przez stronę A na B. Warto jednak podkreślić, że Google nie interpretuje tylko liczby głosów lub linków, ale także strony, które te głosy oddają. Waga głosu i linku zależy od istotności strony, która go oddaje lub od której link wychodzi, a więc ważna strona może pomóc wypromować inną poprzez linkowanie do niej.

PageRank (2)

Sergey Brin, Larry Page (1998), Stanford University, Google

"The anatomy of a large-scale hypertextual Web search engine"
Computer Networks and ISDN Systems. ***30****:107–117*

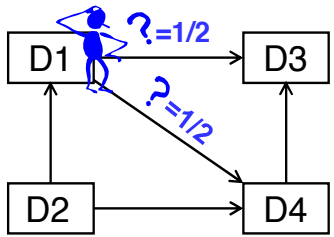
- **Main thesis**: a web page is important if is pointed to by other important web pages
- When one page links to another page it is effectively casting a vote for the other page
- More votes imply greater importance
- Importance of the page that is casting the vote determines the importance of the vote

[7] Podsumujmy najważniejsze zasady, które można wynieść z analizy fragmentu pokazanego na poprzednim slajdzie, pochodzącego z artykułu opublikowanego przez twórców Google'a w czasopiśmie Computer Networks and ISDN Systems. Główna teza głosi, że strona jest tak istotna jak strony, które na nią wskazują. Jeśli jedna strona linkuje do innej, to tak jakby oddawała na nią głos. Oczywiście więcej głosów i linków dochodzących zwiększa ważność danej strony, ale kluczowe jest tu od jakich stron te głosy pochodzą. Ważność strony oddającej głos determinuje bowiem istotność oddawanego przez nią głosu. Ostatecznie dana strona będzie ważna, jeżeli inne ważne strony posiadają wskazania na tę stronę.

PageRank (3) - Random Web Surfer

- PageRank is a numeric value that represents the importance of a page present on the web
- Links between pages form the paths that users can follow

- **Metaphore of a random web surfer**
- Begins his walk at a random page
- At each stage, he selects randomly one of the available outlinks and walks to the page this link indicates
- The number of stages is infinite



- How often will the surfer visit each page?
- **PageRank = the probability of visiting a page in a random walk**

[8] Sam PageRank jest współczynnikiem, który odzwierciedla ważność strony przy wzięciu pod uwagę struktury sieci, a zignorowaniu jej zawartości oraz sposobu korzystania. Interpretacja tego współczynnika odnosi się jednak do analizy korzystania przez użytkowników z sieci traktowanej jako całość. Linki między stronami można bowiem interpretować jako ścieżki, którymi mogą podążać użytkownicy. Wyobraźmy sobie użytkownika poruszającego się po sieci w sposób losowy po linkach między strony. Taki użytkownik, którego możemy nazwać losowym surferem w sieci (ang. random web surfer), rozpoczyna swoją podróż na dowolnej stronie. W każdym etapie losowo wybiera jeden z dostępnych linków wychodzących z tej strony i idzie do strony, do której ten link prowadzi. Taka procedurę, tj. krok od bieżącej strony po jednym z wychodzących od niej linków, jest powtarzana nieskończenie wiele razy. W tym kontekście można się zastanowić jakie jest prawdopodobieństwo, że użytkownik znajdzie się na danej stronie w takiej losowej podróży. Odpowiedź daje właśnie PageRank.

PageRank (4) - Computational Part

- Each page shares its PageRank equally among all the pages that it links to

Second rule

$$PageRank(d_j) = PR(d_j) = \sum_{d_l : d_l \longrightarrow d_j} \frac{PageRank(d_l)}{c(d_l)}$$

PageRank value of page d_j

a number of links outgoing from d_l

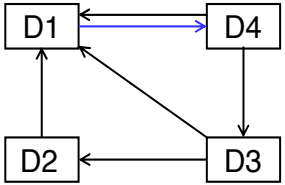
First rule

- Each page derives its PageRank form the importance of pages that link to it (cast their votes for it)

[9] Omówmy teraz aspekty obliczeniowe. Niech punktem wyjścia będzie chęć obliczenia PageRank'a dla strony dj. Zasadą obliczania współczynnika kierują dwie reguły, w których punkt widzenia odnosi się do danej strony traktowanej jako odbiorca i nadawca PageRank's. Jedna reguła mówi że PageRank danej strony bierze się z ważności, tj. PageRank'ów, stron które na nią linkują. Stąd we wzorze suma, w której iterujemy po stronach linkujących do dj i pobieramy ich PageRank. Nie bierzemy jednak całej ważności strony linkującej, bo druga reguła mówi, że każda strona dzieli swój PageRank po równo do stron, na które wskazuje. Stąd w zapisie sumy PageRank każdej strony dl linkującej do dj jest dzielony przez liczby linków z niej wychodzących tak by każdy sąsiad dostał po równo.

PageRank (5) - Stochastic Matrix Representing a Web

- Page D_i corresponds to i-th row and i-th column of matrix M
- M[i,j] = a probability of going from page D_j to page D_i (being currently at page D_j)
- M[i,j] = $1/c(D_j)$, if page D_j has $c(D_j)$ outgoing links (including a link to page D_i)
- M[i,j] = 0, otherwise



fill by columns

M =

D1	D2	D3	D4	
0	1	1/2	1/2	D1
0	0	1/2	0	D2
0	0	0	1/2	D3
1	0	0	0	D4

Example: $PR(D_1) = 0PR(D_1) + PR(D_2) + 1/2 \cdot PR(D_3) + 1/2 \cdot PR(D_4)$

[10] W algorytmach bazujących na strukturze sieci kluczową rolę pełni analiza wybranej macierzy. W przypadku PageRank'a jest to macierz stochastyczna, oznaczona literą M. Jest to macierz kwadratowa, w której i-ty wiersz i i-ta kolumna odpowiada i-tej stronie. W przykładzie na slajdzie mamy do czynienia z siecią składającą się z 4 stron, stąd rozmiar macierzy to 4x4. Macierz taką najlepiej wypełniać kolumnami, bo komórka M[i,j] reprezentuje prawdopodobieństwo przejścia w następnym kroku od strony j z kolumny do strony i z wiersza. Jeśli strona j linkuje do strony i, to w takiej komórce powinno pojawić się 1 przez c(Dj) gdzie c(Dj) jest liczbą linków wychodzących od Dj. Jeśli natomiast takiego linka od strony j do strony i nie ma, to w komórce powinno pojawić się 0. Spójrzmy na kilka przykładów. Żadna stron nie ma linka do samej siebie, stąd 0 na głównej przekątnej. Nie ma też linka od D1 do D2, stąd 0 w komórce M[2,1]. Strona D2 ma tylko 1 link wychodzący i prowadzi on do D1, stąd 1 w komórce M[1,2]. Z kolei D4 ma 2 linki wychodzące, do D1 i D3, stąd M[1,4] = 1/2 oraz M[3,4] = 1/2, a w komórkach M[2,4] oraz M[4,4] są 0. Przeanalizuj, jak wypełniać taką macierz i dlaczego wygodniej wypełniać ją kolumnami. Zwróć uwagę, że liczba niezerowych współczynników w macierzy M odpowiada liczbie 6 linków w naszej sieci. Analiza macierzy stochastycznej pozwala na rozpisanie równania na PageRank każdej ze strony zgodnie ze wzorem omówionym na poprzednim slajdzie. Przykład u dołu slajdu odnosi się do PageRank'a dla strony D1. Można go wyrazić - na postawie pierwszego wiersza w macierzy M - jako sumę PageRank'ów: całkowitego od strony D2, połowy od strony D3 oraz połowy od D4.

PageRank (6) - Stochastic Matrix Representing a Web

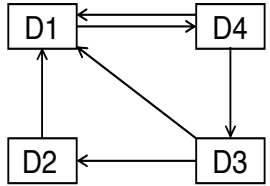
- Let v be a vector of PageRanks to be derived: $v^T = [PR(D_1), PR(D_2), \ldots, PR(D_N)]$
- Solve a system of linear equations v = Mv
  enriched with the normalization contraint $PR(D_1) + PR(D_2) + \ldots + PR(D_N) = 1$

D1	D2	D3	D4	
0	1	1/2	1/2	D1
0	0	1/2	0	D2
0	0	0	1/2	D3
1	0	0	0	D4

$PR(D_1) = PR(D_2) + 1/2 \cdot PR(D_3) + 1/2 \cdot PR(D_4)$
$PR(D_2) = 1/2 \cdot PR(D_3)$
$PR(D_3) = 1/2 \cdot PR(D_4)$
$PR(D_4) = PR(D_1)$
$PR(D_1) + PR(D_2) + PR(D_3) + PR(D_4) = 1$

[11] PageRank każdej strony można więc wyrazić jako sumę odpowiednich proporcji PageRank'ów innych stron. Takie rozważania jak na poprzednim slajdzie dla strony D1 można przeprowadzić dla pozostałych stron: D2, D3 i D4. Przykładowo PageRank dla strony D2 to połowa PageRank'a strony D3. Prowadzi to do sformułowania układu równań v = Mv, gdzie v jest wektorem PageRank'ów, zaś M - macierzą stochastyczną. Istnieją dwa alternatywne podstawowe sposoby obliczenia PageRank'ów na podstawie takiego układu równań. Pierwszy z nich zakłada zastosowanie tradycyjnych technik rozwiązywania układów. Aby uniknąć trywialnego i nieintuicyjnego rozwiązania, w którym wszystkie współczynniki byłyby równe 0, układ jest wzbogacany przez równanie normalizacyjne. My zakładamy, że wszystkie PageRanki mają sumować się do 1. W literaturze można też często spotkać zapis, w którym stała po prawej stronie nie jest równa 1, a liczbie stron w sieci (tu 4). Dla analizowanego przez nas przykładu układ równań zaprezentowano na prawo od macierzy stochastycznej.

PageRank (7) - Example



$PR(D_1) = PR(D_2) + 1/2 \cdot PR(D_3) + 1/2 \cdot PR(D_4)$
$PR(D_2) = 1/2 \cdot PR(D_3)$
$PR(D_3) = 1/2 \cdot PR(D_4)$
$PR(D_4) = PR(D_1)$
$PR(D_1) + PR(D_2) + PR(D_3) + PR(D_4) = 1$

make use of the normalization constraint

$PR(D_4) = PR(D_1)$
$PR(D_3) = 1/2 \cdot PR(D_4) = 1/2 \cdot PR(D_1)$
$PR(D_2) = 1/2 \cdot PR(D_3) = 1/4 \cdot PR(D_1)$

$PR(D_1) + PR(D_2) + PR(D_3) + PR(D_4) = 1$
$PR(D_1) + 1/4 \cdot PR(D_1) + 1/2 \cdot PR(D_1) + PR(D_1) = 1$
$11/4\ PR(D_1) = 1 \rightarrow PR(D_1) = 4/11$

express all variables in function of one specific variable

compute values of all variables

Do these values make sense?

$PR(D_1) = 4/11 = 0.3636$
$PR(D_2) = 1/4 \cdot PR(D_1) = 1/11 = 0.0909$
$PR(D_3) = 1/2 \cdot PR(D_1) = 2/11 = 0.1818$
$PR(D_4) = PR(D_1) = 4/11 = 0.3636$

[12] Aby rozwiązać układ równać zastosujemy prostą, znaną z liceum metodę przez podstawianie. Polega ona na tym, że wszystkie zmienne wyraża się w funkcji jednej z nich i następnie podstawia do równania normalizującego w celu obliczenia jej wartości. To pozwala następnie na obliczenie wartości wszystkich zmiennych. Dla naszego przykładu wyrazimy PageRanki wszystkich stron w funkcji PageRank'a strony D1. Spójrz na ramkę po prawej stronie. Dla D4 nie trzeba nic robić. Dla D3 i D2 jesteśmy w stanie uzyskać odniesienie do D1 w wyniku prostych przekształceń. Podstawiamy więc wyrażenie na PageRanki wszystkich stron w funkcji D1 do równania normalizującego, uzyskując wynik PR(D1) = 4/11 (patrz ramka po prawej). Jesteśmy już w stanie obliczyć PageRanki dla pozostałych stron (patrz ramka na dole). Czy te wyniki mają sens? Najważniejszymi stronami okazały się D1 i D4. W istocie D1 jest wskazywana przez wszystkie pozostałe strony pomimo że D3 i D4 dają jej tylko część swojej mocy. Natomiast D4 jest wskazywana przez D1, ale D1 jest stroną bardzo ważną i przekazuje do D4 całą swoją moc. Najmniej istotną stroną jest D2 - jej PageRank jest czterokrotnie niższy w stosunku do D1 i D4. D2 jest wskazywana tylko przez D3, która nie dość, że nie jest sama bardzo ważna, to oddaje do D2 tylko połowę swojej mocy. Wyniki są więc zgodne z intuicją.

PageRank (8) - Computational Part

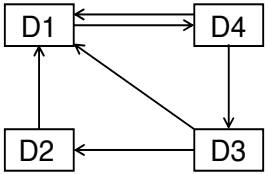
How to solve a system of linear equations v = Mv in a different way?

- Let v be a vector whose i-th element indicates the probability of being at each page Di at some moment in time
- In the initial iteration, assume all probabilities are equal (1/N)
- After one stage, a probability distribution of being at different pages corresponds to the vector Mv
- The probability distribution in an infinite stage corresponds to M(M(M(…M(Mv)…)))

[13] Skupmy się na innej metodzie obliczenia współczynników PageRank. Odniesiemy się do interpretacji praktycznej macierzy stochastycznej M. Załóżmy, że wektor v będzie reprezentował prawdopodobieństwa znalezienia się na wszystkich stronach w danej iteracji. W zerowej iteracji takie prawdopodobieństwa, czyli ważności wszystkich stron, są równe. Przyjmijmy wartość 1/N gdzie N jest liczbą stron w sieci. W każdej kolejnej iteracji i-ta strona przekazuje swoją ważność następnikom, tj. stronom do których posiada linki, a jednocześnie otrzymuje nową ważność od swoich poprzedników, tj. stron, które posiadają do niej linki. Zatem w pierwszej iteracji PageRanki można obliczyć jako Mv (M razy v), w drugiej jako MMv, w trzeciej MMMv, a w nieskończonej wektor v trzeba by pomnożyć przez M nieskończenie wiele razy.

PageRank (9) - Example

The solutions to the equation **v = Mv** can be obtained with an **iterative method** though successive multiplications of the estimates of v by matrix M



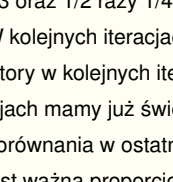
0	**1**	**1/2**	**1/2**
0	0	1/2	0
0	0	0	1/2
1	0	0	0

Multiply a vector from the previous iteration by M

Accurate values are obtained through many iterations

1/4	**1/2**	0.3125	0.3750	0.3437	.	4/11
1/4	1/8	0.0625	0.0625	0.1250	.	1/11
1/4	1/8	0.1250	0.2500	0.1562	.	2/11
1/4	1/4	0.5000	0.3125	0.3750	.	4/11

the same results



A boundary of the probability distribution corresponds to the **principal eigenvector of matrix M** (its elementary values correspond to PageRanks)

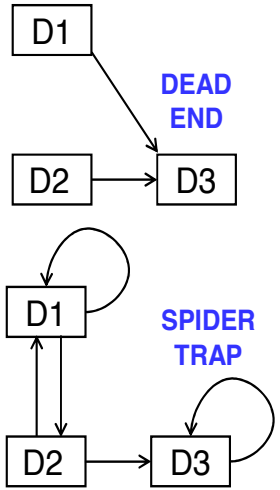
[14] Szczęśliwie nie trzeba takiego mnożenia wykonywać nieskończenie wiele razy, bo z własności matematycznych macierzy M wiemy jaki będzie wynik takiej iteracyjnej metody. Granica rozkładu prawdopodobieństwa, a więc wynik zbiegania wymnożenia wektora v z równymi wartościami przez M nieskończenie wiele razy, to principal eigenvector, a więc wektor własny macierzy M odpowiadający największej wartości własnej. Wektory i wartości własne to pojęcia, które znacie z algebry liniowej. Metodę iteracyjną można jednak efektywnie stosować, bo już po kilku iteracjach mamy bardzo dobre przybliżenie ostatecznego wyniku. Taki proces przedstawiono na slajdzie. W pierwszej kolumnie wszystkie wartości są równe 1/4. W drugiej kolumnie wartość 1/2 dla D1 wynika ze zsumowania 1/4 od D2, 1/2 razy 1/4 od D3 oraz 1/2 razy 1/4 od D4. Z kolei wartość 1/8 dla D2 pochodzi od 1/2 razy 1/4 od D3. W kolejnych iteracjach wyniki powstają na podstawie rezultatów iteracji poprzedniej. Wektory w kolejnych iteracjach stają się coraz bardziej podobne do siebie, a po kilku iteracjach mamy już świetne przybliżenie dokładnego wyniku, który przedstawiono dla porównania w ostatniej kolumnie. Można więc ostatecznie powiedzieć, że strona jest ważna proporcjonalnie do prawdopodobieństwa jej odwiedzenia.

PageRank (10) - Problems Related to the Web Structure

Dead ends and **spider traps** violate the conditions needed for the random walk theorem

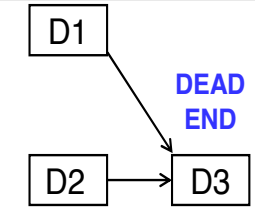
- Pages with no outlinks are **dead ends** for the random surfer (nowhere to go on next step)
- The importance leaks from the web through a page which does not have any outgoing links

- A group of pages is a **spider trap** if there are no links from within the group to outside the group
- Random surfer gets trapped
- Spider trap absorbs the entire importance which is transferred to this group



[15] W przypadku modelowania rzeczywistych struktur sieci występuje wiele problemów, z których dwa podstawowe omówione są na slajdzie. Mogą one prowadzić do zniekształcenia ważności stron i w związku z tym wymagają dedykowanych rozwiązań. Pierwszym problemem jest ślepa uliczka (ang. dead end). Nazywamy tak stronę, która nie posiada następników. Powoduje to, że nie ma ona dokąd przekazać swojej ważności pomimo uprzedniego pobrania jej od innych stron. W przykładzie na slajdzie u góry taką ślepą uliczkę reprezentuje D3. W konsekwencji ważność wszystkich stron jest równa 0. Drugi problem to pułapka pajęcza (ang. spider trap). Nazywa się tak stronę lub grupę stron, która nie posiada linków wychodzących do innych stron, a zawiera linki tylko do siebie lub wewnątrz grupy. W przykładzie na slajdzie na dole taka pułapka jest reprezentowana przez stronę D3. Pobiera ona w sposób bezpośredni (od D2) lub pośredni (od D1) ważność od całej sieci i nie dzieli się nią, przekazując tylko samej sobie. W rezultacie absorbuje ważność całej sieci.

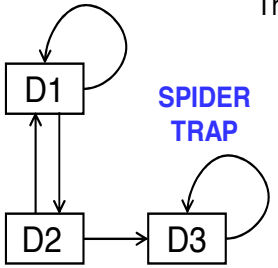
PageRank (11) - Problems Related to the Web Structure



DEAD END

Stochastic matrix			PageRanks in different iterations				
0	0	0	1/3	0	0	.	0
0	0	0	1/3	0	0	.	0
1	1	0	1/3	2/3	0	.	0

The importance leaks from the web through the dead end



SPIDER TRAP

1/2	1/2	0	1/3	1/3	1/4	.	0
1/2	0	0	1/3	1/6	1/6	.	0
0	1/2	1	1/3	1/2	7/12	.	1

The importance is absorbed by the spider trap

The results cannot be interpreted = they do not make much sense

[16] Slajd ilustruje iteracyjny sposób obliczenia PageRank'ów dla sieci z anomaliami. Przedstawiono macierze stochastyczne (przeanalizuj ich wypełnienie wartościami), jak i wartości współczynników w kolejnych iteracjach. W przypadku ślepej uliczki już w drugiej kolumnie jedynie strona D3 ma niezerowy PageRank, a w konsekwencji w kolejnej także jej współczynnik się zeruje. Dla pułapki pajęczej ważność strony D3 w każdej kolejnej iteracji powiększa się kosztem ważności D1 i D2, ostatecznie osiągając wartość 1, pochłaniając całą moc krążącą w sieci. W obydwu przypadkach wyniki nie mogą być interpretowane, bo nawet nie poziomie intuicyjnym nie mają sensu.

PageRank (12) - Google's Revised Formulae

- **Introducing a "tax"**: each page is taxed with a fixed percentage of importance ***q***, the tax is fairly distributed among all pages in the web
- **Random web surfer**: the tax corresponds to a certain probability of a jump to some other part of the web (possibly not connected with the current page)

$$PageRank(d_j) = PR(d_j) = q + (1-q) \sum_{d_l : d_l \longrightarrow d_j} \frac{PageRank(d_l)}{c(d_l)}$$

Damping factor q: usually assumed to be equal 0.15 or 0.20

Solve a system of linear equations **v = q + (1-q)·M·v**

[17] Rozwiązanie problemów ślepej uliczki i pułapki pajęczej, przyjęte już w oryginalnym artykule przez twórców Google, polega na opodatkowaniu każdej strony pewnym procentem jej ważności i rozdystrybuowanie łącznego podatku pomiędzy wszystkie strony w równym procencie. Wprowadzając podatek oznaczony jako q równanie na PageRank ulega modyfikacji. Dana strona na pewno dostanie q, a pozostała część (1-q) jest obliczana zgodnie ze wprowadzonym wcześniej wzorem. Wartość q, którą nazywa się damping factor, zazwyczaj przyjmuje się na poziomie 0.15 lub 0.2. Z punktu widzenia użytkownika poruszającego się po sieci podatek odpowiada pewnemu prawdopodobieństwu skoku ze strony do innej strony bez względu na to, czy łączy je jakiś link czy też nie. W konsekwencji podróży po sieci nie muszą już się odbywać tylko po połączeniach, które istnieją między stronami. Oczywiście układ równań, który trzeba rozwiązań ulega modyfikacji poprzez odpowiednie uwzględnienie w nim wartości stałej q.

PageRank (13) - Example

- Solve a system of linear equations v = q + (1-q)·M·v (without a normalization contraint) using **an iterative method**
- In the below example, we assume q = 0.15

0	1	1/2	1/2
0	0	1/2	0
0	0	0	1/2
1	0	0	0

$PR(D_1) = 0.15 + 0.85 \cdot [PR(D_2) + 1/2 \cdot PR(D_3) + 1/2 \cdot PR(D_4)]$
$PR(D_2) = 0.15 + 0.85 \cdot [1/2 \cdot PR(D_3)]$
$PR(D_3) = 0.15 + 0.85 \cdot [1/2 \cdot PR(D_4)]$
$PR(D_4) = 0.15 + 0.85 \cdot [PR(D_1)]$

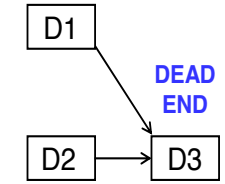
PageRanks in different iterations

							NOW (normalized)	BEFORE (no *q*)
1/4	0.5750	0.6307	0.7707	.	1.4354	1.4354	0.3589	0.3636
1/4	0.2562	0.2589	0.2792	.	0.4611	0.4611	0.1153	0.0909
1/4	0.2562	0.3040	0.4214	.	0.7322	0.7322	0.1831	0.1818
1/4	0.3625	0.6387	0.6861	.	1.3700	1.3700	0.3589	0.3636

not much difference in a web without anomalies

[18] Rozważmy w pierwszej kolejności sieć składającą się z 4 stron (nasz pierwszy przykład), która pozbawiona jest anomalii. Modyfikacja równań na PageRanki poszczególnych stron jest widoczna w ramce na prawo od macierzy stochastycznej. Polega ona na tym, że każda strona dostaje wartość q j(tu 0.15) ako wyjściową, a 1-q, czyli 0.85 jest mnożone przez sumę odpowiednich proporcji PageRank'ów stron, które do niej linkują. W związku z faktem, że wykorzystanie q powoduje, że unikniemy nieinterpretowalnego rozwiązania, w którym wszystkie PageRanki byłyby równe 0, to nie ma potrzeby dodawania równania normalizującego. Rozwiązanie układu równań zostanie dokonane metodą iteracyjną. W dolnej części slajdu zaprezentowano wyniki w kolejnych iteracjach. Startujemy z równych wartościach w pierwszej kolumnie, a wyniki w kolejnych stają się do siebie coraz bardziej podobne. Po normalizacji końcowych rezultatów tak, by wszystkie PageRanki sumowały się do 1, widać, że w sieci która pozbawiona jest anomalii wyniki z uwzględnieniem zastosowania damping factor lub bez niego są do siebie bardzo podobne.

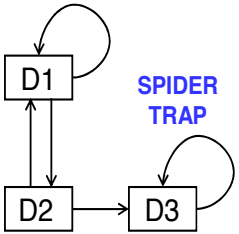
PageRank (14) - Example



PageRanks in different iterations

					Normalized
1/3	0.15 + 0 = 0.150	0.150	.	0.150	0.213
1/3	0.15 + 0 = 0.150	0.150	.	0.150	0.213
1/3	0.15 + 2/3 = 0.716	0.405	.	0.405	0.574

Use **v = q + (1-q)·M·v** in each iteration



1/3	0.433	0.458	.	0.592	0.592	0.181
1/3	0.292	0.334	.	0.380	0.380	0.127
1/3	0.575	0.763	.	2.077	2.077	0.692

The results can be interpreted = they make sense even with the observed anomalies

[19] W przypadku sieci obarczonych anomaliami wyniki po zastosowaniu damping factor nabierają interpretacji. Dla ślepej uliczki - strony D1 i D2 mają jakiś kawałek mocy, którym mogą się dzielić i który jest odnawiany w każdej iteracji. W rezultacie D3 ma największą ważność, co patrząc na sięć jest intuicyjne, a D1 i D2 znacząco niższą. Co jednak istotne, wyniki dla całej sieci nie zerują się. Dla pułapki pajęczej - D3 zagarnia dużo większą moc niż D1 i D2, ale jednak modyfikacja spowodowała, że PageRanki D1 oraz D2 nie zostały wyzerowane. Także i w tym przypadku wyniki są bardziej interpretowalne i zgodne z intuicją.

PageRank (15) - Remarks

Make things easier when computing PageRanks

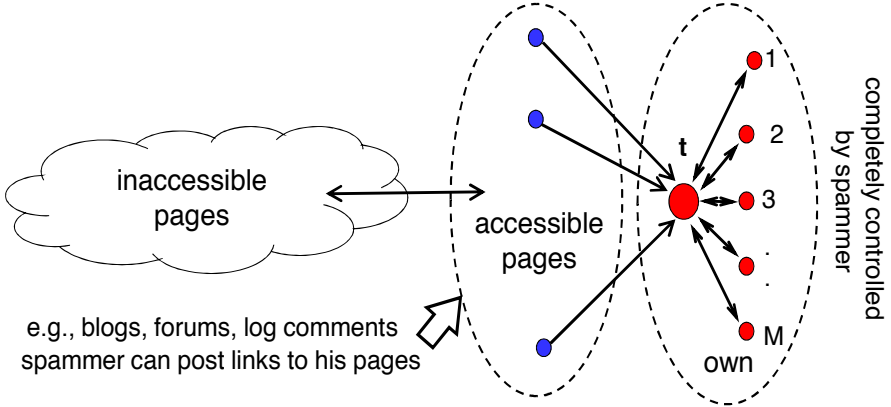
- Predicting PageRanks after few iterations
- Computing local PageRanks on each server
- Simulating simultaneous multiple random web surfers

Major problems / weaknesses

- High values for some particular pages (servers with highly focussed topic)
- Independent from the query/topic
- Vulnerable to link spamming

[20] Proces obliczania wartości PageRank dla większych sieci jest złożony. Dlatego też wprowadza się w nim pewne uproszczenia, zmierzające do estymowania wartości współczynników a nie dokładnego obliczenia ich wartości. Pierwszą z nich jest akceptacja wartości zwróconych po ledwie kilku iteracjach, druga sprowadza się do obliczenia PageRank'ów w małych częściach sieci, a potem interpretacji tylko większych linków np. między serwerami, a trzeci dotyczy przyspieszenia iteracyjnego sposobu, w którym po sieci równolegle puszcza się wielu wirtualnych użytkowników, a wyniki uzyskuje poprzez uśrednienie historii ich odwiedzin. Z wykorzystaniem PageRanków wiąże się też wiele problemów. Pierwszy dotyczy premiowania stron, które wzajemnie do siebie linkują. Zdarza się to choćby w przypadku analizy stron na serwerach dedykowanych ściśle określonemu tematowi. Drugi - wynikający z fundamentalnej zasady działania algorytmu - odnosi się do niezależności od zapytania użytkownika lub jego tematycznych zainteresować. PageRank zależy tylko od struktury sieci. Trzeci nawiązuje do podatności na spamowanie poprzez tworzenie dedykowanych struktur linków, zmierzających do wypromowania ważności niektórych stron.

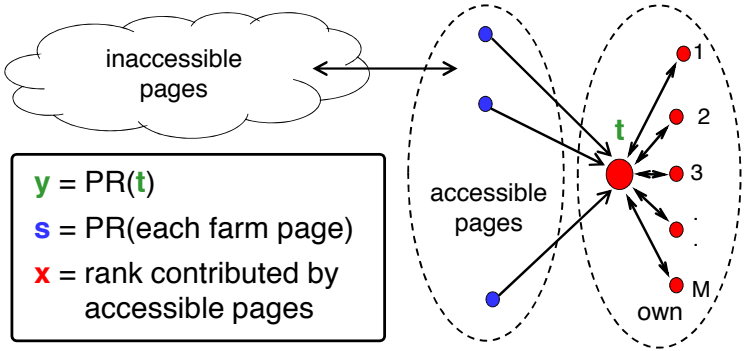
Link Farm (1)



- **Goal**: maximize the page rank of target page t
- Get many links from accessible pages
- Construct “link farm” to get page rank multiplier effect

[21] Jedną z technik spamowania z wykorzystaniem połączeń jest tzw. farma linków. Wyobraźmy sobie, że jesteśmy spamerami i z naszego punktu widzenia istnieją trzy rodzaje stron. Pierwsze to strony, których nie jesteśmy właścicielami i na których nie możemy umieścić żadnych linków. Drugie to strony, jak blogi czy fora, których nie jesteśmy właścicielami, ale na których możemy jednak umieścić odniesienia do naszych stron. I wreszcie trzecia grupa to strony, których jesteśmy właścicielami. Ich struktura jest organizowana w farmie w bardzo specyficzny sposób. Jedna ze stron - nazwijmy ją t - ma być stroną promowaną. Pozostałe M stron mają tylko jeden link wchodzący od strony t oraz jeden link wychodzący do strony t.

Link Farm (2)



y = PR(**t**)
s = PR(each farm page)
x = rank contributed by accessible pages

$y = x + q + (1-q)\cdot M\cdot s$

$s = q + (1-q)\cdot y/M$

$y = x + q + (1-q)\cdot M\cdot[q + (1-q)\cdot y/M]$

$y = x + q + (q-q^2)\cdot M + (1-q)^2\cdot y$

$y - (1-2q+q^2)y = x + q + (q-q^2)\cdot M$

$$y = \frac{x + q + (q-q^2)\cdot M}{2q - q^2}$$

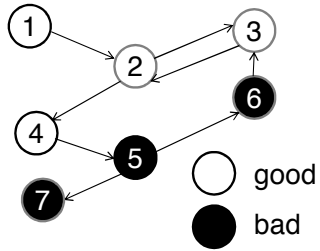
- Multiplier effect
- By making **M** larger, **y** can be make larger

[22] Celem jest uzyskanie jak największej wartości PageRank dla strony t. Oznaczmy go przez y i rozpiszmy go. Składają się na niego x, czyli część dostępna od stron, których nie kontrolujemy, ale na których mogliśmy umieścić linki, q - damping factor, oraz (1-q) pomnożone przez część wynikającą z właściwej części algorytmu zastosowanego do farmy linków. Z tej ostatniej wynika, że każda mała strona w farmie ma swój PageRank - nazwijmy go s - i oddaje go w całości do strony t. Skoro takich stron jest M, to całkowity wkład od takiej farmy wynosi M razy s. Obliczmy więc PageRank dla s - wynika on z q oraz (1-q) pomnożonego przez część wynikającą z właściwej części algorytmu, w której strona t dzieli się swoją M-tą częścią PageRanku y. Podstawiamy s do wyrażenie na y i po kilku przekształceniach uzyskujemy wynik (patrz ramka z prawej strony). Sugeruje on, że y jest wprost proporcjonalny do M. A zatem zwiększając liczbę stron w farmie, można dowolnie zwiększać y. Jest to bez wątpienia duża wada algorytmu.

TrustRank (1) - Algorithm

- Sample a set of “seed pages” from the web
- Oracle (human) identifies the good pages and the spam pages in the seed set
- Propagate trust through links
- Use a threshold value and mark all pages below the trust threshold as spam

Basic principle: approximate isolation
it is rare for a “good” page to point to a “bad” (spam) page

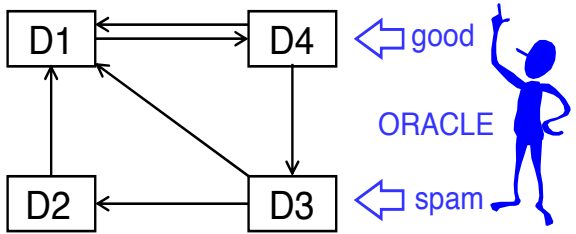


Z. Gyongyi, H. Garcia-Molina and J.Pedersen. Combating Web Spam with TrustRank. Tech. rep., Stanford University, 2004

[23] Omówimy teraz sposoby radzenia są z takimi próbami oszustw. Rozwiązaniem jest tu opracowany na uniwersytecie Stanford algorytm TrustRank. Zakłada on, że człowiek (wyrocznia) oceni w wiarygodny sposób próbkę stron, a ta informacja zostanie rozpropagowana po całej sieci. Skupimy się więc na trzech kluczowych składowych algorytmu: w jaki sposób wybrać próbkę stron (ang. seed) do oceny przez wyrocznię, jakich ocen musi dokonać wyrocznia i jak je uwzględnić w algorytmie, tzn. jak rozpropagować zaufanie lub jego brak po sieci. Warto na wstępie zaznaczyć, że u podstaw TrustRanka leży założenie o przybliżonej izolacji w sieci. Mówi ono, że strony dobre (zaufane) bardzo rzadko linkują do stron złych (niezaufanych). Jest to założenie realistyczne, bo w istocie dobre strony nie mają żadnego interesu, by linkować do stron złych. W przykładowej sieci na slajdzie jedynym takim niepożądanym linkiem jest połączenie między stronami 4 i 5. Warto jednak podkreślić, że pomimo że wszystkie strony w tej sieci są oznaczona jako dobre (białe) lub złe (czarne), to algorytm nie posiada tej wiedzy dla wszystkich stron. Ma on dopiero obliczyć współczynniki zaufania, którą tę dobroć będą aproksymować.

TrustRank (2) - Selection of Seed Pages

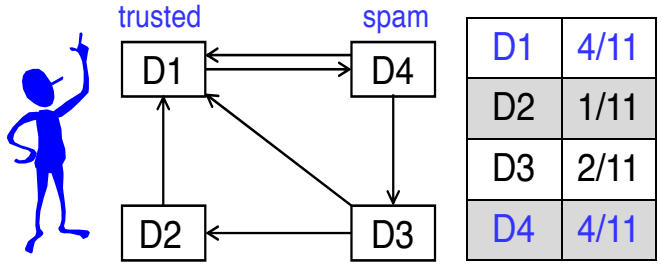
- **Two conflicting considerations:**
- Human has to inspect each seed page, so seed set must be as small as possible (money, expensive task)
- Ensure every “good page” gets adequate TrustRank, so good pages should be reachable from seed set by short paths



[24] Zacznijmy od sposobu wyboru stron, które powinny być ocenione przez wyrocznię. W tym względzie trzeba uwzględnić dwie sprzeczne motywacje. Wyrocznią jest człowiek, które musi ocenić każdą stronę w próbce, a skoro człowiek ma zrealizować jakąś pracę, to trzeba mu za nią zapłacić. Jest to więc zadania kosztowne, a zatem by minimalizować koszty - rozmiar próbki musi być mały. Z drugiej strony, próbka musi być stosunkowo duża tak, by zagwarantować, że zostaną ocenione przez wyrocznię strony w różnych częściach sieci. W konsekwencji dobre strony powinny być stosunkowo łatwo, w kilku krokach, osiągalne ze stron zawartych w takiej próbce.

TrustRank (3) - Selection of Seed Pages with PageRank

- **First idea: PageRank**
- Pick the top k pages by PageRank
- Assume high PageRank pages are close to other highly ranked pages
- We care more about high page rank “good” pages



D1	4/11
D2	1/11
D3	2/11
D4	4/11

[25] Pierwszym pomysłem na wybór ograniczonej liczy stron, które powinny znaleźć się w próbce przedstawionej wyroczni jest zastosowanie algorytmu PageRank. Motywacja jest tu dwutorowa. Po pierwsze, strony z wysokim PageRankiem - zgodnie z zasadą działania algorytmu - są blisko innych wysoko ocenionych stron. Przypisanie im zaufania lub jego braku będzie więc korzystne dla oceny wielu stron znajdujących się wysoko w rankingu wynikającego z analizy struktury sieci. Po drugie, opłaca się je ocenić bezpośrednio, bo strony z wysokim PageRank’iem są przedmiotem naszego największego zainteresowania. W przypadku sieci przedstawionej na slajdzie najwyższy PageRank mają strony D1 i D4. W ogólności dla potrzeb TrustRanka wybrane byłoby k stron z największą wartością PageRank’a, gdzie k jest predefiniowanym rozmiarem próbki.

TrustRank (4) - Seed Selection with Inverse PageRank

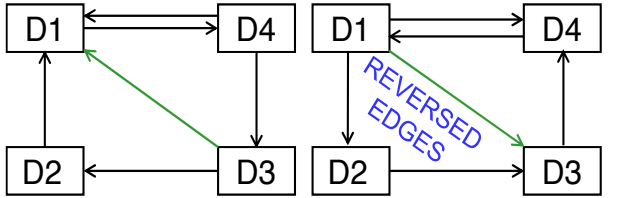
- **Second idea:** Pick the pages with the maximum number of outlinks
- **Recursive:** pick pages that link to pages with many outlinks
- Construct graph G' by reversing each edge in web graph G
- **PageRank in G' is Inverse PageRank in G**
- Pick top k pages by Inverse PageRank

PageRanks

D1	D2	D3	D4
4/11	1/11	2/11	4/11

Inverse PageRanks

D1	D2	D3	D4
3/9	1/9	2/9	3/9



INVERSE STOCHASTIC MATRIX *MI FOR G* = = STOCHASTIC MATRIX FOR G'

0	0	0	1
1/3	0	0	0
1/3	1	0	0
1/3	0	1	0

[26] Drugi - lepszy - pomysł zakłada wykorzystanie algorytmu Inverse PageRank. Podstawowy pomysł zakłada tu wybór stron, które zawierają wiele linków wychodzących jako tych, które mogłyby szybko rozprzestrzenić zaufanie lub jego brak po sieci. Pomysł ten można rozszerzyć, bo można by wybrać strony, które posiadają linki do stron, które posiadają wiele linków wychodzących, itd. To jest właściwa motywacja leżąca u podstaw Inverse PageRank. Algorytm ten opiera się na tej samej zasadzie co PageRank, ale analizuje graf z odwróconymi linkami. U dołu slajdu przedstawiono taką sieć. Przykładowo w oryginalnej sieci znajdował się link od D3 do D1, a w odwróconej od D1 do D3. W prawym dolnym rogu znajduje się macierz stochastyczna dla takiej odwróconej sieci. Nazywamy ją odwrotną macierzą stochastyczną dla sieci oryginalnej. PageRanki obliczone na jej podstawie to odwrócone (ang. inverse) PageRanki dla sieci oryginalnej. W przypadku analizowanej sieci znów największe wartości osiągnięto dla D1 i D4. W ogólności - tak jak na poprzednim slajdzie - wyrocznia powinna ocenić k stron z największą wartością Inverse PageRank.

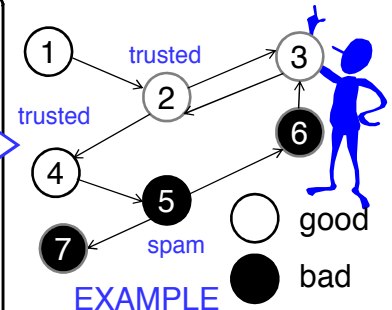
TrustRank (5) - Oracle / Human Quality Tester

s = rank pages according to PR or Inverse PR
σ = **selectSeed(…)** – select **T** best pages
d = [0,0,…,0] – set the **evaluations of the oracle** to zero
for ***i*** = 1 to ***T*** do
    if ***oracle judges the page as trusted*** then d(σ(i)) = 1
d = d / |d| - normalize the vector of oracle's evaluations

Rank pages using Inverse PR
- **s** = [2, 4, 5, 1, 3, 6, 7]

Select T=3 best pages: **σ** = [2, 4, 5]
Invoke T=3 oracle functions
Static score distribution vector
- **d** = [0, 1, 0, 1, 0, 0, 0]

Normalize distribution vector
- d = [0, 1/2, 0, 1/2, 0, 0, 0]



[27] Sformułujmy kroki algorytmu Inverse PageRank ze szczególnym naciskiem na rolę jaką pełni w niej wyrocznia. W pierwszym kroku szeregujemy wszystkie strony zgodnie z algorytmem dedykowanym do wybory próbki stron (może to być PageRank lub jego odwrócona wersja). Spośród tych stron wybiera się ograniczoną liczbę stron najlepszych. Dla wszystkich stron tworzy się wektor ocen d, który wypełniany jest zerami. Zadaniem wyroczni jest ocena dla każdej strony z próbki, czy jest ona zaufana. Jeśli tak, to w wektorze ocen na pozycji odpowiadającej takiej stronie wstawiana jest jedynka. Po wyjściu z pętli wektor ocen jest normalizowany, tak by wartości w nim sumowały się do 1. Dla przykładowej sieci składającej się z 7 stron zastosowano Inverse PageRank. Wektor s zawiera indeksy stron uszeregowane względem Inverse PageRank (najlepsza okazała się stron 2, a najgorsza - 7). Załóżmy, że próbka ma zawierać trzy strony, a zatem trafiają do niej strony 2, 4 i 5. Są one oceniane przez wyrocznię. Strona 2 i 4 są zaklasyfikowane jako zaufane, a strona 5 jako niezaufana (spam). W rezultacie w wektorze, który inicjalizuje działania algorytmu, wartości dodatnie znajdują się tylko dla stron ocenionych jako zaufane przez wyrocznią, a 0 dla stron nieocenionych lub ocenionych jako złe.

TrustRank (6) - Trust Propagation

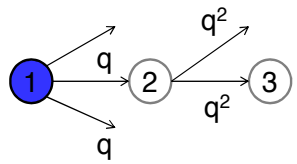
Trust dampening

- We cannot be sure that pages reachable from good seeds are indeed good
- The degree of trust conferred by a trusted page decreases with distance
- q – dampening factor

Trust splitting

- The larger the number of outlinks from a page, the less trust the page author gives each outlink

Can be combined



[28] Ostatnią istotną składową algorytmu TrustRank jest sposób propagacji zaufania po sieci. Powinny nim sterować dwie reguły. Po pierwsze, nie ma pewności, że wszystkie strony osiągalne ze stron zaufanych są dobre. Zaufanie powinno więc maleć wraz ze wzrostem odległości od stron ocenionych jako zaufane. Trend ten powinien mieć charakter potęgowy - w najbliższym sąsiedztwie trafia q zaufania, w sąsiedztwie pośrednim q^2, itd. Po drugie, jeśli strona ma dużo linków wychodzących, to powinna to zaufanie rozprzestrzeniać równo do swoich następników. Oni z kolei powinni rozprzestrzeniać je równo do swoich sąsiadów. Każda strona może też sumować zaufanie pochodzące od różnych stron. Te dwie reguły można połączyć w jedną procedurę. Co więcej, znacie ją już, bo reguły te powinny przypominać Wam fundamentalne założenia algorytmu PageRank.

TrustRank (7) - Formulae and Example

$$PageRank(d_j) = PR(d_j) = q + (1-q) \sum_{d_l : d_l \rightarrow d_j} \frac{PageRank(d_l)}{c(d_l)}$$

Biased PageRank

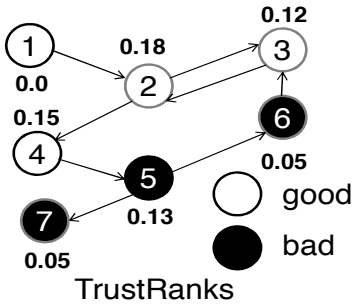
$$TrustRank(d_j) = TR(d_j) = q \cdot d + (1-q) \sum_{d_l : d_l \rightarrow d_j} \frac{TrustRank(d_l)}{c(d_l)}$$

with trusted pages as teleport set

TR = d
for *i* = 1 to *no_iterations* do
TR = q·d + (1- q)·M·TR
return **TR**

TR(1) = q·0.0 + (1-q)·0
TR(2) = q·0.5 + (1-q)·[TR(1)+TR(3)]
TR(3) = q·0.0 + (1-q)·[0.5·TR(2)+TR(6)]
…
TR(5) = q·0.0 + (1-q)·[TR(4)]
…

d = [0, 1/2, 0, 1/2, 0, 0, 0]



[29] TrustRank jest zmodyfikowaną wersją PageRank'a. Można powiedzieć, że wersja ta jest ukierunkowana/skrzywiona (ang. bias). Różnica w obliczeniu tego współczynnika polega tylko i wyłącznie na wymnożeniu wartości damping factor q przez wartość wynikającą z początkowego wektora ocen wyroczni d. Z praktycznego punktu widzenia oznacza to, że losowe kroki w PageRank, które nie prowadzą po istniejących linkach, mogą być dokonywane tylko do stron, które zostały ocenione jako zaufane przez wyrocznię jako zaufane i mają dodatnią wartość w wektorze d. Choć zmiana wydaje się niewielka, to silnie wpływa na końcowe wyniki. Do obliczenia TrustRank'ów jest stosowana metoda iteracyjna. W rozpisce dla poszczególnych stron dla przykładowej sieci można dostrzec, że dla strony D2, q jest mnożone przez dodatnią wartość z wektora d, czyli 1/2, dla D1 - strony nieocenionej - q jest mnożone przez 0, a dla D5 - strony ocenionej jako niezaufana - q jest także mnożone przez 0. Finalne wartości przedstawiono przy odpowiednich wierzchołkach na rysunku w prawym dolnym rogu. Największą wartość zaufania mają strony 2 i 4 ocenione jako zaufane bezpośrednio przez wyrocznię. Do strony 3 też trafia sporo zaufania. Ze względu na strukturę sieci (niepożądany link od 4 do 5) strona 5 ma wysokie zaufanie, ale już strony dalsze - 6 i 7 mają to zaufanie bardzo nikłe.

Google Panda

Google Panda
released in Febraury 2011
named after Navneet Panda

"for the long-term trust of our users and for a better ecosystem for publishers"

- **Human quality testers** rated thousands of websites based on measures of quality, including design, trustworthiness, speed and whether or not they would return to the website
- Google's **machine-learning algorithm** was used to look for similarities between pages people found to be high/low quality
- Better rankings for pages with original content and information: research, reviews, in-depth reports, thoughtful analysis, etc.
- Target pages that aren't necessarily spam but aren't great quality

[30] Wielu osobom wydawało się, że idea TrustRanka zaproponowanego w 2004 roku i zakładająca wykorzystanie ludzkich testerów jest nierealistyczna. Najlepsze zaprzeczenie tych opinii przyszło w 2011 roku, kiedy Google wdrożyło działanie algorytmu Panda, który jest jedną z nakładek na oryginalnego PageRanka. U jego podstaw leżało wykorzystanie testerów, których zadaniem było ocena tysięcy stron pod względem ich jakości, projektu, wiarygodności, szybkości działania oraz chęci powrotu, w sumie ponad 20 cech. Następnie metody uczenia maszynowego wykorzystano, by określić podobieństwo innych stron do stron dobrze lub źle ocenionych przez testerów. Takie wyniki wykorzystano z kolei do modyfikacji ich pozycji w rankingu stron zwracanych przez Google. Okazało się, że bardzo premiowane zostały strony produkujące oryginalne treści (takie jak recenzje, wyniki badań lub reporty) oraz serwisy społecznościowe. Ukarano zaś serwisy, które składały się z wielu podstron, zawierały mnóstwo reklam lub kopiowały treści z innych serwisów. W Polsce algorytm został wdrożony w sierpniu 2011 roku, kilka miesięcy po światowej premierze.

Google Penguin

Google Penguin
released in April 2012

"to decrease search engine rankings of the websites that use forbidden link building techniques"

- Remove sites from the search results that have been trying to buy or "unnaturally acquire" links to their website
- Assigns a negative value to specific links (if you have certain links, your rankings will become lower, not higher)
- Do not violate Google's Webmaster Guidelines

[31] Drugim rozszerzeniem zastosowanym przez Google, a wdrożonym rok później, w 2012, był Penguin. Jego celem było obniżenie pozycji stron pozycjonowanych nieetycznymi metodami (ang. spamdexing). Google opublikowało zalecenia dla twórców stron w postaci Webmaster Guidelines, w których rekomendowało dbałość o jakość linków, ich różnorodność i naturalność, a więc unikanie takich anomalii jak farmy, sieci katalogów czy linki z serwisów niepowiązanych tematycznie, a także unikanie przeoptymalizowania stron i upychania na siłę w ich treści słów kluczowych. Dla stron, które te zasady łamały, ranking ulegał znacznemu obniżeniu, choć warto zaznaczyć, że algorytm działał punktowo na wybrane podstrony, a nie serwisy traktowane całościowo. Od 2012 roku Google wypuściło mnóstwo kolejnych dedykowanych algorytmów, a jednym z ciekawszych jest Hummingbird dedykowany wyszukiwaniu semantycznemu. Warto też poczytać o innych rozszerzeniach algorytmicznych samego PageRanka, jak choćby Topic-Specific PageRank.

HITS (1)

Jon Michael Kleinberg (1999)
IBM, Cornell University

*Kleinberg, J. M. (1999). "Authoritative sources in hyperlinked environment". Journal of the ACM. **46**(5): 604*

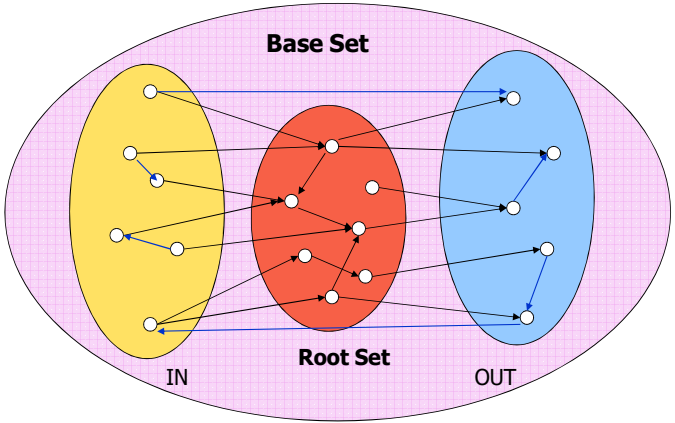
HITS (Hypertext Induced Topic Search) = Hubs & Authorities

- **Main thesis**: a page is a good hub (and therefore deserves a high hub score) if it points to good authorities,
  and a page is a good authority if is pointed to by good hubs
- Query specific: sampling module constructs a set of a few hundred pages relevant to the query (suited for ”broad topic”)
- Propagation model computes the importance coefficients

[32] Mniej sławnym algorytmicznym bratem PageRanka jest algorytm HITS, opracowany przez Kleinberga w 1999 roku. Pomimo że procedura ta nie zrobiła tak zawrotnej kariery, to warto znać kryjące się u jej podstaw idee, bo nawet jeśli dziś rzadko są stosowane w kontekście analizy struktury Internetu, to można - podobnie jak w przypadku PageRank’a - przenieść je na inne pola zastosowań. Choć pełna nazwa HITS to Hypertext Induced Topic Search, to często funkcjonuje on pod nazwą roboczą: Hubs Authorities. Odnosi się ona to dwóch ról, które pełni każda strona zgodnie z założeniami algorytmu, ról koncentratora (ang. hub) oraz autorytetu (ang. authority). HITS wypracowuje wartości dwóch współczynników dla każdej stron, zakładając przy tym bardzo silne relacje między rolą koncentratora i rolą autorytetu. Strona jest bowiem dobrym koncentratorem, czyli drogowskazem w sieci, o ile wskazuje na dobre autorytety. Dla odmiany strona jest dobrym autorytetem, jeśli jest wskazywana przez dobre koncentratory. Mówimy o dwóch rolach, ale wspomnieć też trzeba o dwóch fazach działania algorytmu. Pierwsza, próbkująca - ważna, choć rzadko kojarzona z algorytmem - polega na tym, by wyodrębnić podzbiór stron, dla których współczynniki istotności mają być obliczone. Druga - będąca silnym wyróżnikiem HITSa - to już właściwe obliczenie mocy każdej strony jako autorytetu i koncentratora poprzez propagację.

HITS (2) - Construction of Root and Base Sets

- Scores are calculated with subgraph of the entire web graph
- Construction of a **root set** R_q
- Expansion of a root set to a **base set** S_q by including pages which link to or are linked by the pages from the root set



[33] Pierwsza faza polega na wyodrębnienie podgrafu stron w całej sieci, który jest zależny od zapytania użytkownika. Składa się on zwykle z kilkuset lub kilku tysięcy stron i jest konstruowany w dwóch podetapach. W pierwszym - wyodrębniony jest zbiór początkowy, zbiór korzeń (ang. root set), który identyfikuje się jako czołowe strony z rankingu zwróconego przez wyszukiwarkę dla zapytania użytkownika. W drugim następuje ekspansja - zbiór jest rozszerzony o strony, które linkują do i są linkowane przez strony zawarte w root set'cie. W tak powstałym zbiorze bazowym (ang. base set) zachowywana jest struktura połączeń/linków i to na nim zapuszczany jest właściwy algorytm HITS, a mówiąc precyzyjniej - jego druga faza. Warto podkreślić, że czasami linki pomiędzy stronami, które należą do tej samej domeny są interpretowane jako nawigacyjne i eliminowane z analizy.

HITS (3) - Hubs and Authorities

Two types of pages are distinguished:

- **authorities**: they contain important information for a given query
- **hubs**: they indicate where important information can be found

Recursive definition of different types of pages

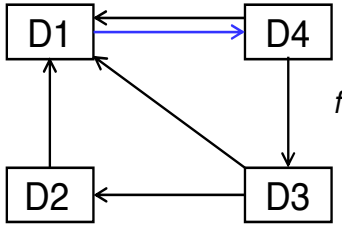
$$h(D_j) = \sum_{D_l \in S_q:\ D_j \longrightarrow D_l} A(D_l) \qquad a(D_j) = \sum_{D_l \in S_q:\ D_l \longrightarrow D_j} H(D_l)$$

- A **good hub** is a page which **links to many good authorities**
- The hub weight is the sum of the authority weights of the authorities pointed to by the hub
- A **good authority** is a page which **is linked by many good hubs**
- The authority weight is the sum of the hub weights that point to this authority

[34] W drugiej fazie HITSa obliczane są współczynniki ważności każdej strony jako autorytetu i koncentratora. Autorytet to strona, która zawiera informację istotną z punktu widzenia zapytania, a koncentrator to strona, która wskazuje, gdzie taką istotną informację można znaleźć. Definicja tych ról jest więc wzajemna, co znajduje odzwierciedlenie w równaniach. Jakość strony jako koncentratora wynika z jakości autorytetów dla stron, które są przez nią linkowane. Dobry koncentrator musi więc wskazywać na wiele dobrych autorytetów. Z kolei jakość strony jako autorytetu jest zdefiniowana jako suma jakości koncentratorów dla stron, które linkują do danej strony. Dobry autorytet musi być więc wskazywany przez wiele dobrych koncentratorów.

HITS (4) - Neighborhood Matrix

- Page D_i corresponds to i-th row and i-th column of matrix L
- M[i,j] = 1, if page D_i links to D_j
- M[i,j] = 0, otherwise



fill by rows ⟹ L =

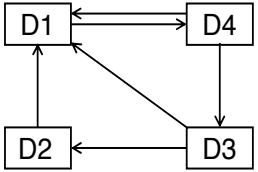
	D1	D2	D3	D4
D1	0	0	0	1
D2	1	0	0	0
D3	1	1	0	0
D4	1	0	1	0

[35] W przypadku PageRank'a kluczowa dla analizy była macierz stochastyczna, a dla HITSa jest to macierz sąsiedztwa, często oznaczana przez L. Szczęśliwie jej interpretacja jest dużo łatwiejsza. Ponownie jest to macierz kwadratowa, gdzie liczba wierszy i kolumn odpowiada liczbie stron w sieci, a i-ty wiersz oraz i-ta kolumna odpowiadają i-tej stronie. Bardziej intuicyjne jest jednak jej wypełnianie. Możliwe wartości to 1 lub 0. Ta pierwsza pojawia się w komórce L[i,j] jeśli strona i-ta (z wiersza) posiada link do strony j-tej (z kolumny). Ta druga (0), jeśli takiego linka nie ma. Macierz sąsiedztwa wygodnie jest więc wypełniać wierszami na podstawie analizy linków wychodzących z danej strony/wierzchołka. Przykładowo, strona D1 posiada tylko jeden link wychodzący, do strony D4, stąd 0 w komórkach [1,1], [1,2] i [1,3] oraz 1 w komórce [1,4]. Z kolei strona D4 posiada linki wychodzące do D1 i D3, co implikuje jedynki w dwóch komórkach w czwartym wierszu. Sumaryczna liczba jedynek (6) odpowiada liczbie linków w sieci.

HITS (5) - Example

L =

0	0	0	1
1	0	0	0
1	1	0	0
1	0	1	0



$$h(D_j) = \sum_{D_l \in S_q:\ D_j \longrightarrow D_l} A(D_l)$$

$$a(D_j) = \sum_{D_l \in S_q:\ D_l \longrightarrow D_j} H(D_l)$$

Let **a** and **h** be vectors of pages' importance in terms of authorities and hubs

L^T =

0	1	1	1
0	0	1	0
0	0	0	1
1	0	0	0

$h(D_1) = 0 \cdot a(D_1) + 0 \cdot a(D_2) + 0 \cdot a(D_3) + 1 \cdot a(D_4)$

$a(D_1) = 0 \cdot h(D_1) + 1 \cdot h(D_2) + 1 \cdot h(D_3) + 1 \cdot h(D_4)$

$\mathbf{h} = \mu \cdot L \cdot \mathbf{a}$ $\quad$ $\mathbf{a} = \nu \cdot L^T \cdot \mathbf{h}$

[36] Niech wektory a oraz h oznaczają odpowiednio wektory ważności stron jako autorytetów oraz koncentratorów, a ich i-ty element odpowiada wartości tych współczynników dla i-tej strony. Z definicji moc strony jako koncentratora jest sumą mocy autorytetów stron, do których linkuje. Dla każdej strony można zapisać odpowiednie równanie na podstawie analizy odpowiedniego wiersza macierzy sąsiedztwa L. Przykładowo, moc koncentratora dla strony pierwszej to suma autorytetów strony pierwszej, drugiej i trzeciej pomnożonych przez 0 oraz autorytet strony czwartej mnożony przez 1 (patrz pierwszy wiersz macierzy L). Dla odmiany moc strony jako autorytetu jest sumą mocy koncentratorów stron, które do nie linkują. Dla każdej strony można zapisać dedykowane równanie na podstawie analizy odpowiedniej kolumny macierzy L, choć dla wygody lepiej operować na wierszach macierzy transponowanej L^T. I tak moc autorytetu dla strony pierwszej to suma koncentratorów strony pierwszej pomnożonej przez 0 oraz autorytetów strony drugiej, trzeciej i czwartej pomnożonych przez 1 (patrz pierwszy wiersz macierzy L^T). Ostatecznie w postaci wektorowej można wyrazić wektor h w funkcji wektora a, jeśli pomnożyć go przez współczynnik skalujący mi oraz macierz L. Z kolei wektor a można wyrazić w funkcji wektora h, jeśli pomnożyć go przez współczynnik skalujący ni oraz macierz L^T.

HITS (6) - Which Matrices to Analyze?

$h = \mu \cdot L \cdot a$

$a = v \cdot L^T \cdot h$

$L^T =$

0	1	1	1
0	0	1	0
0	0	0	1
1	0	0	0

$h = \mu \cdot v \cdot L \cdot L^T \cdot h$

$a = \mu \cdot v \cdot L^T \cdot L \cdot a$

For scalars, the order is not important, unlike for matrices

$L =$

0	0	0	1
1	0	0	0
1	1	0	0
1	0	1	0

1	0	0	0
0	1	1	1
0	1	2	1
0	1	1	2

$= L \cdot L^T$

For **hubs**: $L \cdot L^T$

For **authorities**: $L \cdot L^T$

$0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 1$

$0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 = 3$

3	1	1	0
1	1	0	0
1	0	1	0
0	0	0	1

$= L^T \cdot L$

[37] Z definicji przedstawionych na poprzednim slajdzie wynika, że h zależy od a, natomiast a zależy od h. Podstawmy więc po prawej stronie wzoru na h, wartość a i okaże się, że h można obliczyć na podstawie h, o ile pomnożyć ten wektor przez dwa współczynniki skalujące i macierz LL^T (powstałą z wymnożenia L i L^T). Na podobnej zasadzie podstawmy po prawej stronie wzoru na a, wartość h i okaże się, że a można obliczyć na podstawie a, o ile pomnożyć ten wektor przez dwa współczynniki skalujące i macierz $L^T L$ (powstałą z wymnożenia L^T i L, przy czym ważna jest kolejność mnożenia macierzy). Na slajdzie zaprezentowano macierze LL^T oraz $L^T L$ różniące się kolejnością mnożenia macierzy. Przypomniano także sposób mnożenia macierzy, gdzie wartość w każdej komórce jest sumą iloczynów wartości z odpowiedniego wiersza pierwszej macierzy i kolumny drugiej macierzy. Kluczowy wniosek z tego slajdu jest jednak taki, że dla obliczenia mocy stron koncentratorów h istotna jest analiza macierz LL^T, a dla mocy stron autorytetów a - macierz $L^T L$.

HITS (7) - Principal Eigenvectors

$L \cdot L^T =$

1	0	0	0
0	1	1	1
0	1	2	1
0	1	1	2

$\mathbf{h} = \mu \cdot v \cdot \mathbf{L} \cdot \mathbf{L}^T \cdot \mathbf{h}$

$\mathbf{a} = \mu \cdot v \cdot \mathbf{L}^T \cdot \mathbf{L} \cdot \mathbf{a}$

HUBS ⇨

D1	0.00	0.00
D2	0.73	0.27
D3	1.00	0.37
D4	1.00	0.37

- The hub weight vector is the principal eigenvector of $\mathbf{LL^T}$
- The authority weight vector is the principal eigenvector of $\mathbf{L^TL}$

Results from a calculator for eigenvalues and eigenvectors for $\mathbf{LL^T}$:

```
Real Eigenvalues: { 0.26;  1;  1;  3.73}
Eigenvectors:
 for Eigenvalue 0.26:      [ 0; -2.73; 1; 1]
 for double Eigenvalue 1: [1; 0; 0; 0] or [0; 0; -1; 1]
 for Eigenvalue 3.73:      [ 0; 0.73; 1; 1]
```

Results for $\mathbf{L^TL}$: `for Eigenvalue 3.73: [ 2.73; 1; 1; 0]`

[38] Rozwiązania takich układów równań ponownie można znaleźć na różne sposoby. Pierwszy z nich zakłada bezpośrednie odwołanie się do principal eigenvectora (wektora własnego odpowiadającego największej wartości własnej) odpowiedniej macierzy. Na slajdzie przedstawiono szczegółowe wyniki dla macierzy LL^T istotne w kontekście wartości autorytetów. Największą wartością własną tej macierzy jest 3.73, a odpowiadający jej wektor własny [0, 0.73, 1, 1] stanowi nasze rozwiązanie. Wynik taki można znormalizować, jak pokazano w prawym górnym rogu. Dla macierzy L^TL przedstawiono już tylko wynikowy wektor (patrz dół slajdu).

HITS (8) - Iterative Power Matrix

Can be solved with an iterative power method:

- Assume all values are equal to 1/n
- Successively multiply the vector from the previous iteration by LL^T or L^TL
- Normalize the weights after each iteration to sum up to 1

HUB WEIGHTS

D1	1/4	0.08	0.02	0.01	0.00	.	0.00	0.00
D2	1/4	0.23	0.26	0.27	0.27	.	0.27	0.73
D3	1/4	0.33	0.36	0.36	0.37	.	0.37	1.00
D4	1/4	0.33	0.36	0.36	0.37	.	0.37	1.00

Approximations in the following iterations — Eigenvector

AUTHORITY WEIGHTS

D1	1/4	0.50	0.56	0.57	0.58	.	0.58	2.73
D2	1/4	0.20	0.21	0.21	0.21	.	0.21	1.00
D3	1/4	0.20	0.21	0.21	0.21	.	0.21	1.00
D4	1/4	0.10	0.03	0.01	0.00	.	0.00	0.00

[39] Drugą metodą obliczenia wartości h oraz a jest podejście relaksacyjne (iteracyjne), analogiczne jak w przypadku algorytmu PageRank. Zakładamy, że początkowe wartości mocy autorytetów i koncentratorów są równe. Dla przykładu z 4 stronami, niech wynoszą one 1/4. W kolejnej iteracji są one mnożone przez macierz LL^T w przypadku koncentratorów lub przez macierz L^TL w przypadku autorytetów. Po każdej iteracji wyniki normalizujemy tak, by wartości współczynników sumowały się do 1. Widać wyraźnie, że rezultaty w kolejnych iteracjach stają się do siebie coraz bardziej podobne, a w ostateczności zbiegają do principal eigenvectora. Uzyskany wynik jest dokładnie taki sam jak w przypadku zastosowania kalkulatora wartości i wektorów własnych, który został omówiony na poprzednim slajdzie.

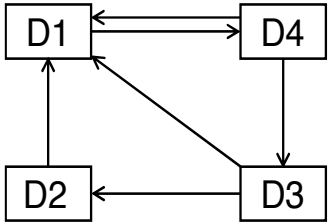
HITS (9) - Example

$L \cdot L^T =$

1	0	0	0
0	1	1	1
0	1	2	1
0	1	1	2

3	1	1	0
1	1	0	0
1	0	1	0
0	0	0	1

$= L^T \cdot L$



HUB WEIGHTS

D1	0.0
D2	0.73
D3	1.0
D4	1.0

⟺

AUTHORITY WEIGHTS

D1	2.73
D2	1.0
D3	1.0
D4	0.0

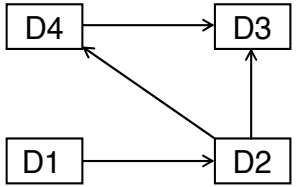
[40] Aby zakończyć analizę przykładu, sprawdźmy, czy uzyskane wyniki mają sens. Najlepszymi koncentratorami okazały się strony D3 oraz D4. D3 wskazuje na D1, który jest najlepszym autorytetem oraz na D2, który jest bardzo dobrym autorytetem. Z kolei D4 wskazuje na D1, tj. najlepszy autorytet oraz D3, czyli także bardzo dobry autorytet. Najgorszym koncentratorem jest D1, bo wskazuje tylko na D4, czyli najgorszy autorytet. Z punktu widzenia autorytetów zdecydowanie najlepszym okazał się D1, bo wskazywany przez dwa najlepsze koncentratory D3 i D4 oraz bardzo dobry koncentrator D2. Najgorszym autorytetem jest D4, który jest wskazywany tylko przez najgorszy koncentrator D1. Wzajemne zależności między koncentratorami i autorytetami znalazły więc odzwierciedlenie w wynikach.

Summary (1)

PageRank	HITS
☺ Probability of visiting a page	☺ Hubs and authorities
☹ Query independent (hooks for peronalization)	☺ Query-specific
☺ Inexpensive at runtime	☹ Expensive at runtime
☺ Scores are calculated with the entire Web graph	☹ Scores are calculated with subgraph of the entire Web
☺ Eigenvector or iterative	☺ Eigenvector or iterative
☹ Prone to manipulations, but these can be detected	

[41] Podsumowując, wykład poświęcony był dwóm najsławniejszym algorytmom w dziedziny eksploracji struktury sieci. PageRank skupia się na pojedynczej roli strony skorelowanej z prawdopodobieństwem znalezienie się na niej przez użytkownika, a HITS wyróżnia dwie uzupełniające się role, która może pełnić każda strona. PageRank nie ma żadnego związku z zapytaniem użytkownika, skupiając się tylko na analizie połączeń, zaś HITS - ze względu na pierwszą fazę - działa na zależnym od zapytania podgrafie całej sieci. Ta ostatnia własność powoduje, że uruchomienie HITSa jest kosztowniejsze niż dla PageRanka. Podstawowe metody obliczenia współczynników ważności stron są jednak podobne dla tych dwóch algorytmów. Wiążą się one z dokładną analizą wektorów własnych, znaną z algebry liniowej, lub wykorzystaniem metody iteracyjnej, która pozwala na dobrą estymację wartości. Kolejny wykład będzie dotyczył analizy użytkowania, w podsumowaniu tego pomoże kilka prostych zadań znajdujących się na dwóch kolejnych slajdach. Zadania te mają pomóc Wam w przygotowaniu do kolokwium zaliczeniowego, które odbędzie (powinno się odbyć) pod koniec semestru.

Summary (2)



I)

$$M = \begin{bmatrix} 0 & ? & 0 & 0 \\ 1 & ? & 0 & 0 \\ 0 & ? & 0 & 1 \\ 0 & ? & 0 & 0 \end{bmatrix}$$

II) $PR(D_3) = ?$

III) Which page has the greatest/least PageRank?

IV) Inverse $PR(D_2) = ?$

V) Which page has the greatest/least Inverse PageRank?

VI) An oracle has evaluated D2 as trusted and D4 as spam? What is the starting vector d for TrustRank?

d = | | | | |

[42] Zadania do samodzielnej realizacji jako powtórka:

I)

II)

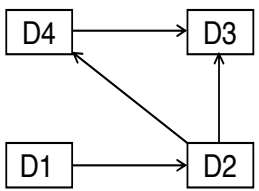
III)

IV)

V)

VI)

Summary (3)



0	0	0	0
1	0	0	0
0	1	0	1
0	1	0	0

$= L^T$

The principal eigenvector of LL^T is [0, 1.618, 0, 1]

The principal eigenvector of L^TL is [0, 0, 1.618, 1]

VII)

$L =$

0	1	0	0
?	?	?	?
0	0	0	0
0	0	1	0

1	0	0	0
0	2	0	1
0	0	0	0
0	1	0	?

$= L \cdot L^T$ **IX)**

VIII) $a(D_4) = ?$
$h(D_4) = ?$

X) Which page has the greatest authority score?

XI) Which page has the greatest hub score?

[43] Zadania do samodzielnej realizacji jako powtórka:
VII)
VIII)
IX)
X)
XI)