Grafy i Zastosowania

10: Zastosowania w sieciach: algorytm PageRank

©Marcin Sydow

Spis zagadnień

- Łańcuch Markowa
  - Macierz przejść
  - Digraf łańcucha Markowa
  - Klasyfikacja stanów
- Zastosowanie: digraf WWW i algorytm PageRank
  - Ranking dokumentów w wyszukiwarkach
  - Podstawy racjonalne analizy linków w liczeniu rankingu
  - Idea PageRank
  - 3 perspektywy: przepływy, losowy internauta i macierze
  - Uproszczony i "realny" PageRank
  - Matematyczne podstawy
  - Obliczanie - metoda Potęgowa
  - Rozszerzenia PageRank

Łańcuchy Markowa

Wyobraźmy sobie następujący proces, który przebiega w czasie.

Mamy **zbiór stanów** V.

W każdym *dyskretnym* momencie czasowym (indeksowanym np. za pomocą liczb naturalnych) $t \in N$ proces ten *jest w pewnym stanie* $v \in V$, w szczególności, w chwili początkowej $t = 0$ system jest w pewnym stanie początkowym $v(0) \in V$.

W następnym momencie $t + 1$ system, zgodnie z tzw. *funkcją przejścia*, *losowo przechodzi* ze stanu $v(t)$ do stanu $v(t + 1)$.

Macierz przejść łańcucha Markowa

W łańcuchu Markowa funkcja przejścia dana jest przez *prawdopodobieństwa przejść* pomiędzy parami stanów w każdym kroku. Ma ona formę tzw. **macierzy przejść** P łańcucha Markowa.

Macierz ta jest kwadratowa, indeksowana stanami i P_{ij} jest prawdopodobieństwem przejścia ze stanu i do stanu j w dowolnym kroku.

Macierz ta ma następującą własność: suma elementów dowolnego wiersza wynosi 1 (suma prawdopodobieństw wszystkich możliwości przejścia z danego stanu). Własność ta nazywana jest *wierszową stochastycznością* macierzy.

przykład

Digraf łańcucha Markowa

Zauważmy, że łańcuch Markowa o zbiorze stanów V, i macierzy przejścia P można naturalnie reprezentować jako digraf $D = (V, E)$, gdzie zbiór wierzchołków to zbiór stanów, a łuk $(i, j) \in E \Leftrightarrow$ gdy $p_{ij} > 0$ (można przejść ze stanu i do stanu j).

Wartości prawdopodobieństw p_{ij} można wtedy reprezentować jako wagi krawędzi (i, j).

Obserwacja:
Macierz przejść P łańcucha Markowa stanowi macierz sąsiedztwa odpowiadającego mu digrafu D

przykład

Macierz przejść, c.d.

Twierdzenie:
Jeśli rozkład prawdopodobieństwa bycia łańcucha Markowa o macierzy przejść P w poszczególnych stanach w momencie t jest dany wektorem X_t to rozkład prawdopodobieństwa X_{t+1} w momencie $t+1$ dany jest wzorem:

$$X_{t+1} = P^T \cdot X_t$$

(P^T oznacza operację transpozycji macierzy P, zakładamy, że wektory X są kolumnami)

(dowód: wynika z własności sumowania prawdopodobieństw wykluczających się)

przykład

Wniosek:
rozkład prawdopobieństwa po k krokach dany jest wzorem:

$$X_{t+k} = (P^T)^k \cdot X_t$$

(dowód: wielokrotne zastosowanie twierdzenia)

Klasyfikacja stanów

Stan v jest:

- **powracający** $\Leftrightarrow$ będąc w nim w momencie t prawdopodobieństwo ponownego bycia w nim w pewnym czasie $t' > t$ wynosi 1 (na pewno wrócimy)
- **chwilowy** $\Leftrightarrow$ nie jest powracający
- **pochłaniający** $\Leftrightarrow$ prawdopodobieństwo przejścia w jednym kroku z v do innego stanu wynosi 0
- **okresowy** o okresie $1 < \tau \in N \Leftrightarrow$ powrócić do stanu v można tylko po liczbie kroków będącej wielokrotnością τ
- **ergodyczny** $\Leftrightarrow$ jest powracający i nie jest okresowy

Uwaga: powyższe kategorie nie są wzajemnie wykluczające się (np. pochłaniający jest powracający, etc.)
przykład

Ergodyczny Łańcuch Markowa

Łańcuch Markowa nazywamy **ergodycznym** $\Leftrightarrow$ każdy jego stan jest ergodyczny.

Twierdzenie:
Ergodyczny łańcuch Markowa ma **rozkład stacjonarny** czyli istnieje graniczny rozkład prawdopodobieństwa bycia w poszczególnych stanach gdy czas dąży do nieskończoności. Nie zależy to od stanu początkowego.

Twierdzenie:
Łańcuch jest ergodyczny $\Leftrightarrow$ odpowiadający mu digraf jest silnie spójny i największy wspólny dzielnik długości cykli w grafie wynosi 1.
(dowód: proste analogie między digrafem a łańcuchem Markowa)

przykład

Przykład: Zastosowanie w wyszukiwarkach WWW (PageRank)

Moduły wyszukiwarki

- Moduł zbierający (ang. Crawler)
  - podążaj po linkach i ściągaj dokumenty
- Repozytorium
  - składuj ściągnięte dokumenty - trwałość, dostęp
- Indeks
  - zapisz które słowo występuje w jakim dokumencie
- **System Rankingowy**
  - **jakie informacje dobrze pasują do zapytania użytkownika?**
  - **jakie informacje są wartościowe same w sobie?**
- Moduł prezentacji
  - znajdź dobrą formę wizualizacji wyników
- Obsługa
  - obsłuż zapytania, znajdź strony, wyświetl wyniki

Szukanie igły w stogu siana - Ranking

Przeciętne zapytanie: **tysiące zwróconych** dokumentów

Możliwości użytkownika: **kilkanaście obejrzanych** dokumentów

Szukanie igły w stogu siana - Ranking

Przeciętne zapytanie: **tysiące zwróconych** dokumentów

Możliwości użytkownika: **kilkanaście obejrzanych** dokumentów

Jak wybrać **na początek listy te kilkanaście najlepszych** spośród tysięcy?

Szukanie igły w stogu siana - Ranking

Przeciętne zapytanie: **tysiące zwróconych** dokumentów

Możliwości użytkownika: **kilkanaście obejrzanych** dokumentów

Jak wybrać **na początek listy te kilkanaście najlepszych** spośród tysięcy?

Rozwiązaniem jest: System Rankingowy

Systemy rankingowe istniały od lat w IR, ale nie były idealne w przypadku WWW
(rewolucja wyszukiwarkowa AD 1998)

Ranking

Najpilniej strzeżone tajemnice wyszukiwarek (decydują o **jakości wyników**)

Dokumentowi przyporządkowana jest wartość (ang. score) i wyniki są posortowane po tej wartości

Wiele składowych:

- analiza tekstu (zawartość, URL, meta, ...)
- analiza tekstu odnośników (ang. anchor text)
- **analiza struktury linków**
- analiza logów, ruchu internetowego, ...

Tekst a ranking

- statystyki (np. tf-idf)
- pozycja w tekście
- pozycja w kontekście (URL, meta, title, anchor, etc.)
- meta-znaczniki
- znaczniki prezentacji (rozmiar, pogrubienie nagłówek)

WWW - problemy z tekstem

Klasyczne, tekstowe techniki IR sprawiają problemy w przypadku WWW:

- Problem **braku samo-opisu**
  (np. zapytanie: "japoński producent samochodów")
- Problem różnorodności
- Problem nierównej jakości
- Zaszumienie, błędy, etc
- Tekst - **łatwy do spamowania**

WWW - rozwiązanie problemów IR

WWW z jednej strony **stwarza problemy** dla klasycznego IR. Z drugiej strony, **stwarza możliwości** ich obejścia dzięki istnieniu dodatkowych źródeł informacji:

- społeczny aspekt publikowania w WWW (linki)
- tekst odnośników (ang. anchor text)

To są mocne narzędzia:

- ominięcie problemu braku samo-opisu
- dokumenty nietekstowe
- dokumenty o nieznanych formatach
- dokumenty nieściągnięte

Dodatkowo: nazwa hosta, domeny, pliku, głębokość ścieżki, ilośc dokumentów na hoście, ...

Linki są użyteczną informacją

Skupmy się na wykorzystaniu analizy linków grafu WWW do automatycznego obliczania rankingu dokumentów WWW

Struktura linków w grafie WWW może zostać wykorzystana do automatycznego obliczania "ważności" (lub jakości) dokumentów, **niezależnie** od kontekstu zapytania.

Taki składnik rankingu (niezależny od zapytania) nazywamy **statycznym**

Ważną cechą linkowego składnika rankingu danego dokumentu jest to, że pochodzi **spoza** tego dokumentu.

Społeczny aspekt hiperlinków

Podstawowa obserwacja:

Zamieszczenie linku z dokumentu p do dokumentu q może być odebrane jako informacja, że podmiot tworzący dokument p uważa dokument q za **wartościowy** (skoro wybrał go do wskazania spośród miliardów innych)

W ten sposób sami twórcy dokumentów WWW są w ukryty sposób "zaprzęgnięci" do oceny dokumentów WWW.

Pojedynczy link nie jest może bardzo wartościową informacją, ale mechanizm ten zastosowany w skali masowej zaczyna działać...

“Nepotyzm”

Problem stanowi tzw. “nepotyzm” linków, czyli tworzenie linków wskazujących dokumenty będące pod kontrolą tego samego podmiotu, który tworzy link. Nie każdy nepotyczny link jest tworzony w złej woli, ale oczywiście takie linki powinny być inaczej (słabiej) uwzględniane

Główny problem polega na niemożliwości pewnego ustalenia czy link tworzony jest przez ten sam podmiot, który kontroluje wskazywany dokument. WWW nie zawiera mechanizmu pozwalającego to sprawdzić.

Reakcja na "nepotyzm"

Typową heurystyką jest traktowanie całego hosta (lub poddomeny) jako przestrzeni kontrolowanej przez pojedynczy podmiot (autora)

W praktyce stosuje się kilka metod uwzględniania "nepotyzmu" opartego na hostach, np:

- ważenie linków w ten sposób, że z każdym hostem związana jest ograniczona wielkość, która jest rozdzielana (np. po równo) pomiędzy wszystkie wychodzące z niego linki
- **ignorowanie** linków wewnątrz hosta (lub poddomeny) przy obliczaniu rankingu opartego na analizie linków

Linki a ważność dokumentu: zliczanie linków wchodzących

Skoro każdy link z dokumentu p do dokumentu q może być traktowany jako informacja, że dokument q jest "wartościowy" (w oczach autora dokumentu p) najprościej byłoby oceniać "ważność" lub "jakość" dokumentu docelowego q poprzez **zliczanie linków wchodzących do q** (ang. backlink count).

Im wyższy stopień wchodzący dokumentu q (backlink count) tym dokument może być ważniejszy (skoro wielu autorów wskazuje ten dokument)

Jest to analogiczne do "głosowania" dokumentów na inne dokumenty (każdy link to jeden głos)

To rozwiązanie ma poważną wadę:

Linki a ważność dokumentu: zliczanie linków wchodzących

Skoro każdy link z dokumentu p do dokumentu q może być traktowany jako informacja, że dokument q jest "wartościowy" (w oczach autora dokumentu p) najprościej byłoby oceniać "ważność" lub "jakość" dokumentu docelowego q poprzez **zliczanie linków wchodzących do q** (ang. backlink count).

Im wyższy stopień wchodzący dokumentu q (backlink count) tym dokument może być ważniejszy (skoro wielu autorów wskazuje ten dokument)

Jest to analogiczne do "głosowania" dokumentów na inne dokumenty (każdy link to jeden głos)

To rozwiązanie ma poważną wadę:

Jest **bardzo podatne na celowe** manipulacje (ang. Search Engine Spam)

Ulepszony pomysł

Przy traktowaniu każdego linku jako równoważnego głosu i jednocześnie braku naturalnego mechanizmu w WWW pozwalającego identyfikować "nepotyzm" każdy podmiot może stworzyć **dowolną** ilość dokumentów zawierających linki do wybranego dokumentu będącego pod kontrolą tego samego podmiotu.

Ulepszenie: nie ważna jest **ilość** linków tylko ich **jakość**

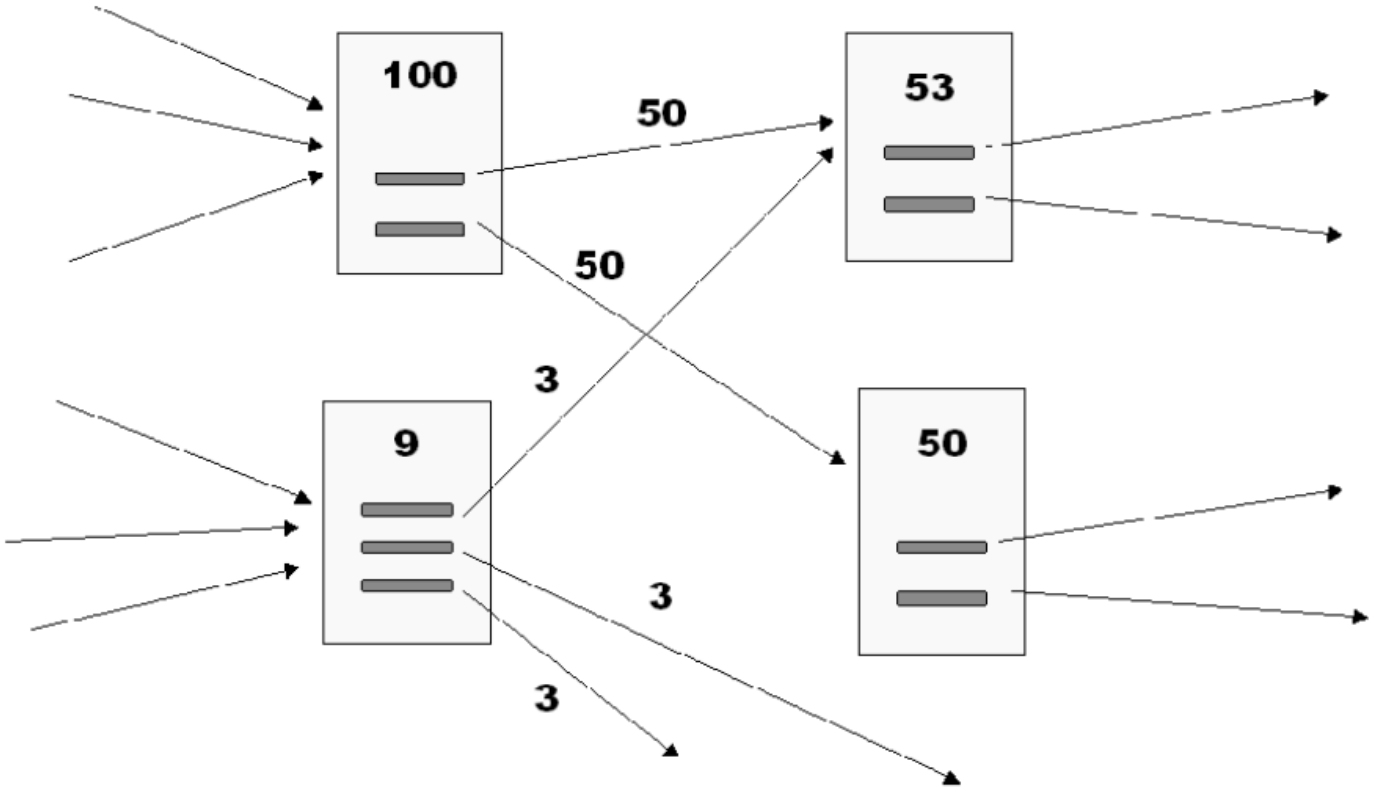
Analogia z głosowaniem: przy zliczaniu głosów uwzględnia się "reputację" głosujących.

Jeden link z bardzo ważnej strony może znaczyć dużo więcej niż 1000 linków z mało ważnych stron.

Za tą ideą (wziętą z m.in. analizy cytowań bibliograficznych) poszli twórcy algorytmu PageRank (ok 1998 roku)

Idea w uproszczeniu - przepływ "wartości" stron

- każda strona ma pewną wartość
- każda strona "głosuje" (poprzez linki) na inne strony
- o wartości strony decyduje wartość stron na nią głosujących



PageRank - uproszczone sformułowanie (perspektywa 1)

Interesuje nas przepływ przez graf WWW taki, że:

- Wartość przepływu sumuje się do 1
- to co wpływa = temu co wypływa (a'la prawo Kirchoffa 1)
- przepływ rozdziela się po równo

Daje to następujące równania:

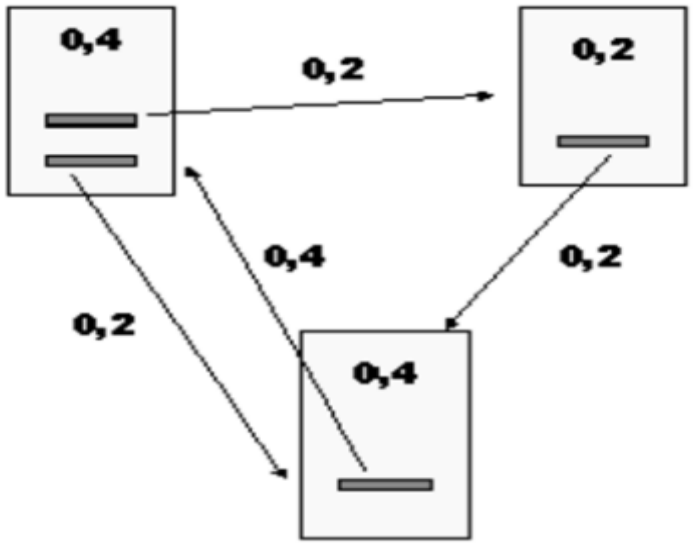
$$\sum_{d \in V} R(d) = 1, \tag{1}$$

$$\sum_{p \in IN(d)} R(p,d) = \sum_{q \in OUT(d)} R(d,q), \tag{2}$$

$$R(q) = \sum_{p \in IN(q)} R(p)/outDeg(p), \tag{3}$$

PageRank to wartość tego przepływu R(d) dla każdego dokumentu d

Przykład dla bardzo prostego grafu



Rysunek: (Jedyny) spełniający warunki przepływ przez przykładowy graf

Perspektywa 2: metafora "losowego internauty" (ang.Random Surfer)

Równoważne zdefiniowanie uproszczonego PageRanku:
Wyobraźmy sobie nieśmiertelnego internautę, który w każdej jednostce czasu przebywa na jakiejś stronie WWW i powtarza następującą akcję:

- wybiera (jednorodnie) losowo wychodzący link i podąża nim na następną stronę

Definition

Uproszczony PageRank dla strony d to graniczna średnia część jednostek czasu spędzonych na stronie d, dla wyżej opisanego procesu, przy czasie dążącym do nieskończoności.

Matematyk powie: "o ile granica istnieje..." I słusznie.

Perspektywa 3 - w języku macierzy

- G(V,E) - rozważany graf
- P - macierz sąsiedztwa G(V, E) zmodyfikowana w ten sposób, że każdy wiersz i jest podzielony przez $outDeg(d_i)$.

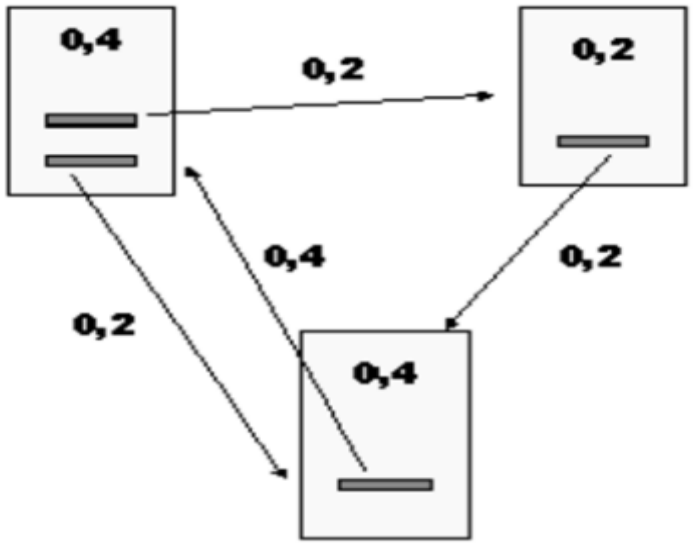
Oba poprzednie sformułowania PageRanku można wyrazić następująco:

Definition

PageRank
to wektor R będący punktem stałym przekształcenia liniowego P^T:

$$R = P^T R \tag{4}$$

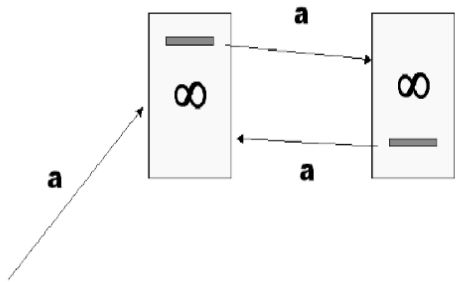
Przykład grafu i (jedyne) rozwiązanie



$$R = P^T R \tag{5}$$

$$\begin{pmatrix} 0.4 \\ 0.2 \\ 0.4 \end{pmatrix} = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}^T \begin{pmatrix} 0.4 \\ 0.2 \\ 0.4 \end{pmatrix} \tag{6}$$

Problemy z uproszczonym PageRankiem



Rysunek: "czarne dziury" (ang. rank sinks)

Problemy:

- Każdy maksymalny podgraf właściwy nie posiadający linków wychodzących pochłania cały PageRank w grafie.
- dokumenty nielinkowane otrzymają zerową wartość.

Jak poprawić uproszczony PageRank?

Jak poprawić uproszczony PageRank?

- łączymy każdy dokument bez wychodzących linków z każdym dokumentem

Jak poprawić uproszczony PageRank?

- łączymy każdy dokument bez wychodzących linków z każdym dokumentem
- dodajemy sztuczne linki pomiędzy wszystkimi pozostałymi parami dokumentów. Są one ważone ułamkowym współczynnikiem $0 < d < 1$ zwanym *decay factor*

Jak poprawić uproszczony PageRank?

- łączymy każdy dokument bez wychodzących linków z każdym dokumentem
- dodajemy sztuczne linki pomiędzy wszystkimi pozostałymi parami dokumentów. Są one ważone ułamkowym współczynnikiem $0 < d < 1$ zwanym *decay factor*
- "prawdziwe" linki ważymy wartością (1 - d)

Jak poprawić uproszczony PageRank?

- łączymy każdy dokument bez wychodzących linków z każdym dokumentem
- dodajemy sztuczne linki pomiędzy wszystkimi pozostałymi parami dokumentów. Są one ważone ułamkowym współczynnikiem $0 < d < 1$ zwanym *decay factor*
- "prawdziwe" linki ważymy wartością (1 - d)

Powyższe sprawi, że w macierzy przejść P każdy wiersz będzie się sumował do 1. (przedtem niektóre wiersze były zerowe) Macierz taka nazywa się *stochastyczna* i istnieje dla niej jednoznaczne rozwiązanie równania

$$R = P^T R \tag{7}$$

Rozwiązanie to jest głównym wektorem własnym tej macierzy.

Przykład na macierzach: (decay factor: 0.1)

$$P_0 = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$P_1 = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 1/60 & 28/60 & 28/60 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 28/60 & 28/60 \\ 1/60 & 1/60 & 1/60 & 28/60 & 1/60 & 28/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 55/60 & 1/60 \end{pmatrix}$$

Poprawiony PageRank w języku losowego internauty...

W każdej jednostce czasu losowy internauta przebywający na stronie s dokonuje następującej akcji:

- jeśli s zawiera linki wyjściowe:
  - z prawdopodobieństwem (1 - d) wybiera (jednorodnie) losowo link wychodzący z danej strony i nim podąża.
  - z prawdopodobieństwem d skacze do dowolnej losowo wybranej strony
- jeśli strona s nie ma linków wychodzących - z prawdopodobieństwem 1 skacze do losowo wybranej strony.

Poprawiony PageRank w języku losowego internauty...

W każdej jednostce czasu losowy internauta przebywający na stronie s dokonuje następującej akcji:

- jeśli s zawiera linki wyjściowe:
  - z prawdopodobieństwem (1 - d) wybiera (jednorodnie) losowo link wychodzący z danej strony i nim podąża.
  - z prawdopodobieństwem d skacze do dowolnej losowo wybranej strony
- jeśli strona s nie ma linków wychodzących - z prawdopodobieństwem 1 skacze do losowo wybranej strony.

Definition

PageRank jest to rozkład stacjonarny zdefiniowanego powyżej nieredukowalnego i acyklicznego łańcucha Markowa

(rozkład ten określa graniczne prawdopodobieństwo bycia internauty na poszczególnych stronach)

...i w języku przepływów

- Uproszczony PageRank:

$$R(p) = \sum_{i \in IN(p)} R(i)/outDeg(i),$$

...i w języku przepływów

- Uproszczony PageRank:

$$R(p) = \sum_{i \in IN(p)} R(i)/outDeg(i),$$

- Dodanie sztucznych linków (uspójnienie):

$$R(p) = (1-d) \sum_{i \in IN(p)} \frac{R(i)}{outDeg(i)} + d \cdot v(p)$$

...i w języku przepływów

- Uproszczony PageRank:

$$R(p) = \sum_{i \in IN(p)} R(i)/outDeg(i),$$

- Dodanie sztucznych linków (uspójnienie):

$$R(p) = (1-d) \sum_{i \in IN(p)} \frac{R(i)}{outDeg(i)} + d \cdot v(p)$$

- Uwzględnienie "przymusowego" skoku z dokumentów bez linków wychodzących:

$$R(p) = (1-d) \sum_{i \in IN(p)} \frac{R(i)}{outDeg(i)} + d \cdot v(p) + (1-d)v(p) \sum_{i \in ZEROS} R(i)$$

Obliczanie PageRank z matematycznego punktu widzenia

$$R = P^T R \tag{8}$$

Z punktu widzenia matematyki, znalezienie wektora R jest łatwe.
Znajdowanie głównego wektora własnego jest równoważne rozwiązaniu układu równań liniowych.

Obliczanie PageRank w praktyce...

Czy można obliczyć PageRank rozwiązując układ równań?

Obliczanie PageRank w praktyce...

Czy można obliczyć PageRank rozwiązując układ równań?
Problemem jest rozmiar zadania.

Obliczanie PageRank w praktyce...

Czy można obliczyć PageRank rozwiązując układ równań?
Problemem jest rozmiar zadania.
Dla przykładu: załóżmy, że ilość dokumentów w grafie to 85M.

- Czas obliczeń: rozwiązywanie układu n równań ma złożoność $\Omega(n^2)$
- Rozmiar macierzy: $7,2P \times 4B = 28PB$ (!)

Co najmniej z tych powodów należy szukać specjalnych metod.

Obejście problemu czasu obliczeń

Metoda Potęgowa: Pozwala szybko obliczyć główny wektor własny macierzy w iteracjach, z teoretycznie dowolną precyzją:

1. $R_0 = v(p)$
2. $i = 0$
3. $R_{i+1} = P^T \cdot R_i$
4. i++
5. if $((|R_{i+1} - R_i| < threshold)$ OR $(i > max))$: stop
6. else: goto 3

Stawiamy pytanie: dla jakich macierzy P metoda potęgowa zbiega i daje jednoznaczny wektor R?

Warunki stosowalności metody potęgowej

Theorem

Metoda potęgowa zbiega do jednoznacznego rozwiązania R równania:

$$R = P^T R \tag{9}$$

jeśli stochastyczna macierz P jest nieredukowalna (odpowiada grafowi silnie spójnemu) i acykliczna. Wtedy, R to główny wektor własny tej macierzy.

- Graf silnie spójny: istnieje ścieżka między każdymi dwoma wierzchołkami
- Macierz acykliczna - odpowiada grafowi, w którym największy wspólny dzielnik długości wszystkich nietrywialnych cykli wynosi 1

Zauważmy, że dodanie sztucznych linków uczyniło graf silnie spójnym i acyklicznym.

Metoda potęgowa jest więc matematycznie poprawna

Obejście problemu rozmiaru macierzy

- Macierz P jest bardzo duża.
- Oryginalna macierz P_0 (odpowiadająca uproszczonemu PageRankowi) jest jednak rzadka - zawiera "prawie same zera". Zmodyfikowane macierze P_1 i P_2 wprawdzie nie są już rzadkie, ale zmiany w stosunku do P_0 dadzą się wyrazić poprzez pojedyncze wektory
- W praktyce oznacza to, że informacje o strukturze grafu przechowuje się w postaci list sąsiedztwa.
- Rozmiar listy sąsiedztwa dla grafu G(V,E) to O(|E|).
- Pojedyncza iteracja metody potęgowej jest zdominowana przez jednokrotny przegląd listy sąsiedztwa

Szybkość metody potęgowej

W praktyce więc, pojedyncza iteracja dla grafu G(V,E) ma złożoność liniową (O(|V|))
Co ciekawe, ilość iteracji nie zależy silnie od |V|.
Ilość iteracji zależy od:

- współczynnika *decay factor*
- progu błędu t

Przy ustalonym progu błędu ilość iteracji metody potęgowej zależy od drugiej głównej wartości własnej macierzy P.
Można pokazać, że druga główna wartość własna P to wlaśnie (1 - d).
Wartość residuum zbiega do zera tak jak $(1-d)^n$
W praktyce ilość iteracji nie przekracza 100 dla zupełnie zadowalającej precyzji.

Usprawnienia obliczeniowe PageRank

Ze względu na rolę algorytmu PageRank i pokrewnych algorytmów w wyszukiwarkach oraz wielkość danych na których one pracują intensywnie badano usprawnienia związane z praktycznym ich obliczaniem:

- efektywne obliczanie w ograniczonej pamięci (podział grafu)
- adaptacyjne obliczanie (wykorzystanie niejednorodnej zbieżności na posczególnych węzłach grafu)
- wykorzystanie matematycznych własności równania PageRank (druga wartość własna)
- wykorzystanie blokowej struktury grafu WWW do równoległego obliczania PageRank
- przyspieszone obliczanie po niewielkich modyfikacjach grafu WWW

Problem "zwisających linków"

Nie jest możliwe posiadanie grafu całego WWW - ma się jedynie dostęp do jego części uzyskanej w procesie crawlowania. W związku z tym, problem stanowi "brzeg" crawla - ta część dokumentów, do których odkryto linki, ale których nie zdążono ściągnąć. Linki takie nazywa się "zwisającymi" (ang. dangling). Niestety, brzeg crawla rośnie w czasie i jego rozmiar zwykle przekracza rozmiar ściągniętego grafu, dla dużych crawli.
Aby to obejść przed liczeniem PageRank można usunąć w i iteracjach (ok. 5) zwisające linki aby dodać je z powrotem do grafu w ostatnich i iteracjach metody potęgowej.

Status algorytmu PageRank

PageRank jest opatentowany w USA:

- Method for node ranking in a linked database
  Inventor: Lawrence Page
  Assignee: The Board of Trustees of the Leland Stanford Junior University
  US Patent 7,058,628
  Granted June 6, 2006
  Filed July 2, 2001
- Filed January 9, 1998 and granted September 4, 2001:
  Method for node ranking in a linked database
- Filed July 6, 2001, and granted September 28, 2004:
  Method for scoring documents in a linked database

Znaczenie PageRank

Nowatorski w 1998 roku algorytm PageRank zrewolucjonizował rynek wyszukiwarek.

Niewielka, dysponująca niewielkim budżetem wyszukiwarka zaczęła skutecznie rywalizować z ówczesnymi gigantami dzięki pomysłowemu algorytmowi, który potrafił efektywnie i trafnie automatycznie porządkować wyniki wyszukiwania.

Obecnie, znaczenie klasycznego algorytmu PageRank w porządkowaniu wyników zmniejszyło się, gdyż wynaleziono techniki "oszukiwania" i jego (mimo, że z założenia należy do bardziej odpornych na manipulacje). Aktualna wersja używana przez wyszukiwarkę, w której powstał nie jest oczywiście publicznie znana i jest zaledwie jednym z wielu czynników uwzględnianych przy obliczaniu rankingu.

Rozszerzenia PageRank

Ze względu na swoje znaczenie historyczne, praktyczne zastosowania i ciekawe własności matematyczne algorytm PageRank doczekał się ogromnej ilości wariantów i rozszerzeń.

Do ważnych rozszerzeń należą m.in.:

- wersje personalizowane
- Topic-sensitive PageRank (czyli zależny od kontekstu zapytania)
- Trust-Rank, i Anti-TrustRank, (zastosowania w zwalczaniu spamu)
- rozmaite wersje rozszerzające model losowego internauty

Personalizacja

Klasyczna wersja PageRank pozwala na prostą i efektywną obliczeniowo "personalizację" za pomocą odpowiedniej modyfikacji "wektora ucieczki". W klasycznej wersji jest on jednorodny, ale już w pierwszej, oryginalnej publikacji na temat PageRank rozważano tę możliwość.

Personalizacja w tym wypadku polega na odpowiednim zwiększeniu prawdopodobieństw przejścia do dokumentów "bardziej interesujących" kosztem zmniejszenia pozostałych prawdopodobieństw.

Pomysł rozwiązania problemu skalowalności masowej personalizacji wektorów ucieczki jest zaprezentowany w: G.Jeh et al. "Scaling Personalized Web Search", WWW Conference 2003 (best paper award)

Topic-Sensitive PageRank

Klasyczny PageRank jest "statyczny" tzn. niewrażliwy na kontekst zapytania przychodzącego do wyszukiwarki.

Zaproponowano wersję "kontekstową" - wrażliwą na temat zapytania. Ranking dokumentu zależy wtedy nie tylko od struktury linków ale i od tematu zapytania.

T.Haveliwala "Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search", WWW Conference 2002

TSPR - Idea

W klasycznym PageRanku liczy się (przed przetwarzaniem zapytania) **1 wektor** rankingu dla wszystkich dokumentów w kolekcji WWW.

W wersji Topic-Sensitive zaproponowano policzenie **wielu wektorów** (oryginalnie 16) - każdy z innym wektorem ucieczki - specjalnie dobranym do wybranej, "reprezentacyjnej" grupy tematycznej. Oryginalnie zaproponowano wykorzystanie 16 głównych kategorii ODP (Open Directory Project).

Przy obliczaniu rankingu dokumentu **w kontekście** zapytania q, bierze się kombinację liniową 16 rankingów, gdzie współczynniki wyrażają "bliskość" zapytania q do każdego z 16 składników tematycznych.

W pracy wykazano eksperymentalnie efektywność tego podejścia i jego przewagę jakościową nad klasycznym

Rozszerzanie modelu losowego internauty

Innym kierunkiem rozszerzania klasycznego algorytmu PageRank jest rozszerzanie bazowego modelu losowego inernauty poprzez dozwalanie na więcej akcji (niż wybór linku i skok do losowej strony)

Na przykład, oprócz 2 w/w akcji bardzo często wykonywaną akcją w przeglądarkach jest użycie klawisza "wstecz" (ang. "back-button").

Okazuje się, że da się tak zmodyfikować klasyczny model, żeby rozwiązanie było matematycznie zbieżne i zarazem efektywnie obliczalne na dużych grafach (mimo, że wynikowy proces nie jest już łańcuchem Markowa). Algorytm (RBS) pracuje na rzeczywistych grafach WWW. ("Random Surfer with back-step", M.Sydow, WWW Conference 2004, (oraz Fundamenta Informaticae, 2005))

Podsumowanie

- Łańcuch Markowa
  - Macierz przejść
  - Klasyfikacja stanów
  - Digraf łańcucha Markowa
- Zastosowanie: digraf WWW i algorytm PageRank
  - Ranking dokumentów w wyszukiwarkach
  - Podstawy racjonalne analizy linków w liczeniu rankingu
  - Idea PageRank
  - 3 perspektywy: przepływy, losowy internauta i macierze
  - Uproszczony i "realny" PageRank
  - Matematyczne podstawy
  - Obliczanie - metoda Potęgowa
  - Rozszerzenia PageRank

Przykładowe pytania/ćwiczenia/zadania

- reprezentuj dany łańcuch Markowa digrafem lub macierzą
- oblicz rozkład stanów po k krokach (k małe)
- Dlaczego ranking jest tak ważny w wyszukiwarkach?
- Ranking statyczny i dynamiczny
- Racjonalne podstawy analizy linków w obliczaniu rankingu
- Nepotyzm i jego neutralizowanie
- Idea PageRank
- 3 perspektywy
- Uproszczony PageRank i jego wady
- "Realny" PageRank
- Równanie PageRank i warunki jego rozwiązalności
- Algorytm Potęgowy obliczania PageRank
- Problem "zwisających" linków
- Rozszerzenia PageRank

Dziękuję za uwagę