Algorytmy skalowalnego przetwarzania danych — projekt 4

dr Piotr Przymus, mgr Mikołaj Fejzer, dr Krzysztof Rykaczewski

1 czerwca 2020

Spis treści

1 Teoria

1.1 Głosowanie poprzez linki

Idea:

- Głosowanie na strony poprzez linki.
- **Strona, która ma więcej linków jest ważniejsza !**
- Liczymy linki odsyłające do strony czy linki wychodzące ze strony ?

Myślimy o linkach odsyłających do strony jako o głosach:

- www.stanford.edu ma ok 23 400 linków odsyłających do tej strony,
- jakas-malo-popularna-domena.pl ma np. 1 link odsyłający.

Czy wszystkie linki są równie ważne ?

- Linki z **ważnych** stron powinny mieć większą wagę !

Jest to rekurencyjne pytanie.

- Każdy link jest liczony jako głos proporcjonalnie do ważności źródłowej strony WWW.
- Jeżeli strona j z poziomem ważności r_j ma n wychodzących linków, to każdy z tych linków dostanie r_j/n głosów.
- Poziom ważność strony j jest ustalany poprzez sumę głosów przychodzących.

1.2 PageRank: model przepływowy

- Głosy z ważnych stron są warte więcej.
- Strona jest ważna, jeśli inne ważne strony wskazują na nią.
- Zdefiniujmy stopień strony r_j dla strony j jako

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}, \tag{1}$$

  gdzie d_i to stopień wyjściowy węzła i.

1.3 PageRank: reprezentacja macierzowa

Stochastyczna macierz incydencji M

- Niech strona i ma d_i wychodzących linków.
- Jeśli $i \to j$, wtedy $M_{ji} = \frac{1}{d_i}$ w przeciwnym wypadku $M_{ji} = 0$.
- M jest macierzą kolumnowo stochastyczną.
  - Tzn. kolumny sumują się do 1.

Wektor stopni r: wektor w którym trzymamy stopień wszystkich stron (każda strona ma swoją pozycję).

- r_j stopień ważności strony i,
- $\sum_i r_i = 1$.

Równania przepływu mogą zostać zapisane jako $r = M \cdot r$, gdzie

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}. \tag{2}$$

1.4 Reprezentacja grafów

Mamy kilka możliwych reprezentacji grafów w tym:

- **(rzadką transponowaną) macierz "sąsiedztwa"**: wypełniamy ją zerami (jeśli dwa wierzchołki nie są połączone krawędzią) i jedynkami (jeśli dwa wierzchołki są połączone); Złożoność pamięciowa: $\mathcal{O}(V^2)$
- **lista incydencji**: należy utworzyć listę dla każdego wierzchołka v, w której przechowujemy zbiór wierzchołków połączonych krawędzią z v; Złożoność pamięciowa: $\mathcal{O}(V+E)$
- **lista krawędzi** (zobacz TGF); Złożoność pamięciowa: $\mathcal{O}(E)$

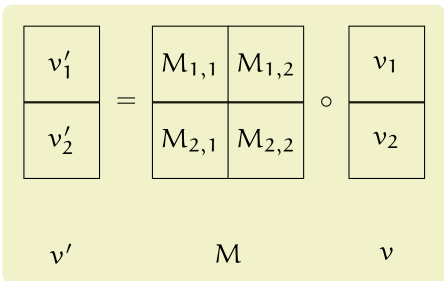
Na ich podstawie możemy stworzyć **macierz przejścia**, czyli taką, która pokazuje przepływ w algorytmie PageRank.

1.5 Podział na fragmenty

W pierwszym zadaniu należy skorzystać z postaci rzadkiej macierzy.

Do kolejnych zadań należy napisać (zwykły, nie musi być w MapReduce — ale może) skrypt/program do generowania postaci zoptymalizowanej (ewentualnie również bloki, zależnie od zadania):

`rowNumber outDegree neighbourVerticesInTheSameBlock`



Rysunek 1: Podział macierzy przejścia.

W zoptymalizowanej wersji PageRank będziemy korzystać z takiego podziału. Zobacz dla przykładu 2, bloki te mają postać (podział na 2).

Wówczas kod potrzebnego do obliczeń bloku można policzyć z numeru wiersza, kolumny i długości bloku (ew. ilości bloków).

W tym kontekście zoptymalizowana znaczy, że nastąpił podział macierzy na bloki, przez co można fragmentami update-ować elementy wektora PageRank. Tyle nam wystarczy w każdym bloku, żeby wykonać obliczenia według rysunku 6.

Nie jest to reprezentacja najmniejsza.

2 Zadania PageRank

Celem jest kod pozwalający obliczyć uproszczony model PageRank. Niezbędne wyjaśnienia znajdują się w tym dokumencie oraz:

- dołączonych slajdach do wykładu "Algorytmy eksploracji i przetwarzania masywnych zbiorów danych". Wykład 7 i 8,
- w rozdziałach 5.1 oraz 5.2 książki *"Mining of Massive Datasets"*, Jure Leskovec, Anand Rajaraman, Jeff Ullman.

Demonstracja oparta o przykładowe grafy umieszczone na końcu tego zestawu.

2.1 Błądzenie losowe (30 pkt)

- Reprezentacja grafu to macierz rzadka z projektu 3. Na potrzeby zadania możemy założyć, że wektor rang mieści się w pamięci.
- Przygotować kod liczący (na zamieszczonych poniżej przykładach grafów) podstawowy algorytm PageRank $v' = M \cdot v$, czyli *de facto* wykonujący mnożenie macierz-wektor.
- W zadaniu należy policzyć 50 iteracji algorytmu.

2.2 PageRank with Taxation (30 pkt)

Przygotować odpowiedni kod i zademonstrować działanie podstawowej wersji PageRank:

- Należy napisać skrypt, który tworzy plik reprezentujący połączenia w sieci.
- Przygotować kod pozwalający policzyć PageRank with Taxation, tzn. korzystając ze wzoru $v' = \beta \cdot M \cdot v + (1 - \beta) \cdot e/n$.
- Policzyć dla $\beta = 0.8$.
- Wykonać 50 iteracji algorytmu.

2.3 Poprawiony PageRank (40 pkt)

- Przygotować i zademonstrować zoptymalizowaną wersję PageRank (patrz rysunek 2). Należy napisać skrypt, który podzieli macierz na bloki w oszczędnej reprezentacji, stosując technikę opisaną w rozdziale 5.2.1 w/w książki.

- Przygotować i zademonstrować *Topic-Sensitive PageRank*. Zobacz przykład 2 z wyróżnionym zbiorem S.

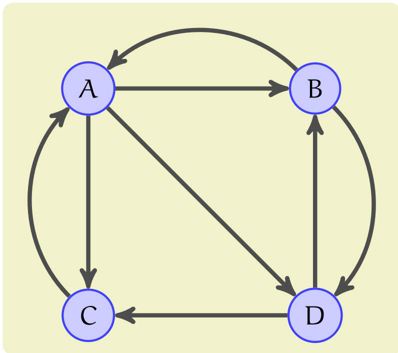
$$\underset{v'}{\begin{bmatrix} v'_1 \\ v'_2 \\ v'_3 \\ v'_4 \end{bmatrix}} = \underset{M}{\begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \\ M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} \\ M_{4,1} & M_{4,2} & M_{4,3} & M_{4,4} \end{bmatrix}} \circ \underset{v}{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}}$$

Rysunek 2: PageRank

3 Przykłady

Demonstracja działa algorytmów, przykłady na podstawie zadań i podanie wyników po 10 iteracjach. Parametr $\beta = 0.8$.

3.1 Przykład 1



Rysunek 3: Przykład 1.

Macierz sąsiedztwa:

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

Macierz przejścia:

$$\begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Widać, że jest to macierz stochastyczna. Tylko macierz z tego przykładu spełnia założenia o zbieżności procesu Markowa.

Rzadka macierz przejścia

```
0 1 0.5
0 2 1
1 0 0.33333333
1 3 0.5
2 0 0.33333333
2 3 0.5
3 0 0.33333333
3 1 0.5
```

Należy ją czytać w ten sposób: "prawdopodobieństwo przejścia z wierzchołka 0 do wierzchołka 1 wynosi 0.5, prawdopodobieństwo. . . "

Z zestawu 2 wiemy jak taką macierz pomnożyć przez wektor rang.

Oszczędne przedstawienie w postaci listy incydencji

Od	Do	Opis
0	1 2 3	A→B,A→C,A→D
1	0 3	B→A, B→D
2	0	C→A
3	1 2	D→B,D→C

Z takiej reprezentacji też moglibyśmy wygenerować rzadką postać macierzy przejścia. Jej specyfikacja to

```
outDegree
neighbours
```

Miejsce w kolejności ustala id wierzchołka.

Jest to tylko przykładowa reprezentacja macierzy w zoptymalizowanej formie. Można też wykorzystać inne reprezentacje podobnej postaci (patrz książka, rozdział 5.2), np.

```
rowNumber outDegree neighbours
```

Błądzenie losowe

```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.375, 0.208333, 0.208333, 0.208333 ]
[ 0.34375, 0.21875, 0.21875, 0.21875 ]
[ 0.332031, 0.222656, 0.222656, 0.222656 ]
[ 0.333252, 0.222249, 0.222249, 0.222249 ]
[ 0.333336, 0.222221, 0.222221, 0.222221 ]
[ 0.333333, 0.222222, 0.222222, 0.222222 ]
[ 0.333333, 0.222222, 0.222222, 0.222222 ]
[ 0.333333, 0.222222, 0.222222, 0.222222 ]
[ 0.333333, 0.222222, 0.222222, 0.222222 ]
...
[ 3/9, 2/9, 2/9, 2/9 ]
```

Korzystając z wolframalpha.com sprawdzamy:

```
eigenvectors {{0,1/2,1,0},{1/3,0,0,1/2},{1/3,0,0,1/2},{1/3,1/2,0,0}}

v_1 = (3/2, 1, 1, 1)
v_2 = (-3, 1, 1, 1)
v_3 = (-3/2, 1, -1/2, 1)
```

Można sprawdzić, że `v_1` odpowiada wartości własnej 1. Po znormalizowaniu dostajemy `(3/9, 2/9, 2/9, 2/9)`.

PageRank with Taxation

```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.35, 0.216667, 0.216667, 0.216667 ]
[ 0.326, 0.224667, 0.224667, 0.224667 ]
[ 0.321136, 0.226288, 0.226288, 0.226288 ]
[ 0.321421, 0.226193, 0.226193, 0.226193 ]
[ 0.321429, 0.226190, 0.226190, 0.226190 ]
[ 0.321429, 0.226190, 0.226190, 0.226190 ]
[ 0.321429, 0.226190, 0.226190, 0.226190 ]
[ 0.321429, 0.226190, 0.226190, 0.226190 ]
[ 0.321429, 0.226190, 0.226190, 0.226190 ]
...
[ 27/84, 19/84, 19/84, 19/84 ]
```

Teoria mówi, że to wektor własny odpowiadający największej wartości własnej. Policzmy macierz Google'a:

```
8/10
* {{0,1/2,1,0},
   {1/3,0,0,1/2},
   {1/3,0,0,1/2},
   {1/3,1/2,0,0}}
+ (1-8/10)
* {{1/4,1/4,1/4,1/4},
   {1/4,1/4,1/4,1/4},
   {1/4,1/4,1/4,1/4},
   {1/4,1/4,1/4,1/4}}

{{1/20, 9/20, 17/20, 1/20},
 {19/60, 1/20, 1/20, 9/20},
 {19/60, 1/20, 1/20, 9/20},
 {19/60, 9/20, 1/20, 1/20}}
```

Korzystając z wolframalpha.com sprawdzamy:

```
eigenvectors {{1/20, 9/20, 17/20, 1/20}, {19/60, 1/20, 1/20, 9/20},
{19/60, 1/20, 1/20, 9/20}, {19/60, 9/20, 1/20, 1/20}}

v_1 = (27/19, 1, 1, 1)
v_2 = (-3, 1, 1, 1)
v_3 = (-3/2, 1, -1/2, 1)
```

Po znormalizowaniu `v_1` mamy `(27/84, 19/84, 19/84, 19/84)`. Odpowiada on wartości własnej 1.

Topic-Sensitive PageRank

Zaufany jest węzeł C.

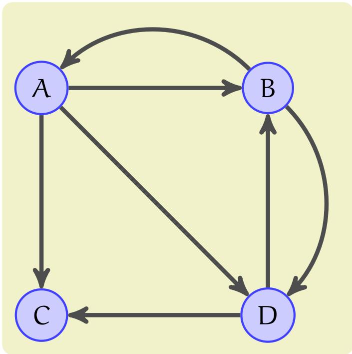
```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.3     , 0.166667, 0.366667, 0.166667 ]
[ 0.336   , 0.154667, 0.354667, 0.154667 ]
[ 0.343296, 0.152235, 0.352235, 0.152235 ]
[ 0.342868, 0.152377, 0.352377, 0.152377 ]
[ 0.342857, 0.152381, 0.352381, 0.152381 ]
[ 0.342857, 0.152381, 0.352381, 0.152381 ]
[ 0.342857, 0.152381, 0.352381, 0.152381 ]
[ 0.342857, 0.152381, 0.352381, 0.152381 ]
[ 0.342857, 0.152381, 0.352381, 0.152381 ]
...
[ 36/105, 16/105, 37/105, 16/105 ]
```

Sprawdzamy na wolframalpha.com

```
8/10 * {{0,1/2,1,0},{1/3,0,0,1/2},{1/3,0,0,1/2},{1/3,1/2,0,0}}
+ (1 - 8/10) * {{0,0,0,0},{0,0,0,0},{1,1,1,1},{0,0,0,0}}

{{0, 2/5, 4/5, 0}, {4/15, 0, 0, 2/5}, {7/15, 1/5, 1/5, 3/5}, {4/15, 2/5, 0, 0}}

eigenvectors {{0, 2/5, 4/5, 0}, {4/15, 0, 0, 2/5},
{7/15, 1/5, 1/5, 3/5}, {4/15, 2/5, 0, 0}}

v_1 = {9/4, 1, 37/16, 1}
```

Po normalizacji `v_1` okazuje się być szukanym wektorem.

3.2 Przykład 2



Rysunek 4: Przykład 2.

Macierz sąsiedztwa

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

Macierz przejścia

$$\begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Nie jest to macierz stochastyczna! Mamy wyciek (ślepy zaułek, *dead end*).

Rzadka macierz przejścia

```
0 1 0.5
1 0 0.33333333
1 3 0.5
2 0 0.33333333
2 3 0.5
3 0 0.33333333
3 1 0.5
```

Oszczędne przedstawienie

Od	Do	Opis
0	1 2 3	A→B,A→C,A→D
1	0 3	B→A, B→D
3	1 2	D→B,D→C

Podział na fragmenty

Blok 11

```
0   3   1
1   2   0
```

Blok 12

```
3   2   1
```

Blok 21

```
0   3   2   3
1   2   3
```

Blok 22

```

3   2   2
```

Błądzenie losowe (zwykły)

```
[ 0.25, 0.25, 0.25, 0.25]
[ 0.125   , 0.208333, 0.208333, 0.208333 ]
[ 0.072917, 0.107639, 0.107639, 0.107639 ]
[ 0.028501, 0.041522, 0.041522, 0.041522 ]
[ 0.008035, 0.011710, 0.011710, 0.011710 ]
[ 0.001651, 0.002406, 0.002406, 0.002406 ]
[ 0.000247, 0.000360, 0.000360, 0.000360 ]
[ 2.697633e-05, 3.931603e-05, 3.931603e-05, 3.931603e-05]
[ 2.145051e-06, 3.126255e-06, 3.126255e-06, 3.126255e-06]
[ 1.242937e-07, 1.811491e-07, 1.811491e-07, 1.811491e-07]
...
[ 0, 0, 0, 0 ]
```

PageRank with Taxation

```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.15, 0.216667, 0.216667, 0.216667 ]
[ 0.120667, 0.157111, 0.157111, 0.157111 ]
[ 0.105240, 0.134043, 0.134043, 0.134043 ]
[ 0.101800, 0.129033, 0.129033, 0.129033 ]
[ 0.101382, 0.128422, 0.128422, 0.128422 ]
[ 0.101353, 0.128380, 0.128380, 0.128380 ]
[ 0.101351, 0.128378, 0.128378, 0.128378 ]
[ 0.101351, 0.128378, 0.128378, 0.128378 ]
[ 0.101351, 0.128378, 0.128378, 0.128378 ]
...
[ 15/148, 19/148, 19/148, 19/148 ]
```

Zauważmy, że macierz przejścia nadal nie jest stochastyczna.

```
8/10 * {{0,1/2,0,0},{1/3,0,0,1/2},{1/3,0,0,1/2},{1/3,1/2,0,0}}
+ (1-8/10) * {{1/4,1/4,1/4,1/4},{1/4,1/4,1/4,1/4},{1/4,1/4,1/4,1/4},{1/4,1/4,1/4,1/4}}

{{1/20, 9/20, 1/20, 1/20}, {19/60, 1/20, 1/20, 9/20},
{19/60, 1/20, 1/20, 9/20}, {19/60, 9/20, 1/20, 1/20}}
```

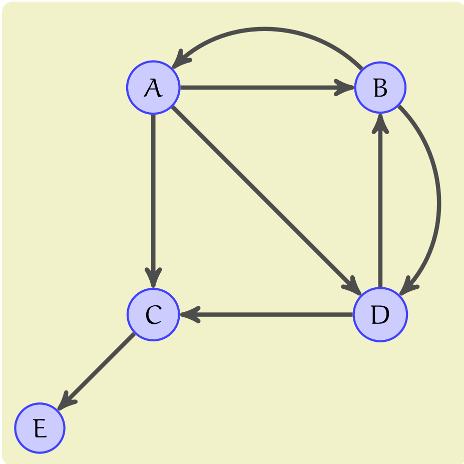
Te kolumny, które sumowały się do 1 nadal to robią, a te, które miały same zera się, mają teraz sumę $1 - \frac{8}{10} = \frac{1}{5}$.

Topic-Sensitive PageRank

Zaufany jest węzeł A.

```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.3     , 0.166667, 0.166667, 0.166667 ]
[ 0.258667, 0.129778, 0.129778, 0.129778 ]
[ 0.246213, 0.112441, 0.112441, 0.112441 ]
[ 0.243587, 0.108608, 0.108608, 0.108608 ]
[ 0.243266, 0.108142, 0.108142, 0.108142 ]
[ 0.243244, 0.108109, 0.108109, 0.108109 ]
[ 0.243243, 0.108108, 0.108108, 0.108108 ]
[ 0.243243, 0.108108, 0.108108, 0.108108 ]
[ 0.243243, 0.108108, 0.108108, 0.108108 ]
...
[ 9/37, 4/37, 4/37, 4/37 ]
```

3.3 Przykład 3



Rysunek 5: Przykład 3.

Macierz sąsiedztwa

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

Macierz przejścia

$$\begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Nie jest to macierz stochastyczna! Mamy wyciek.

Rzadka macierz przejścia

```
0 1 0.5
1 0 0.33333333
1 3 0.5
2 0 0.33333333
2 3 0.5
3 0 0.33333333
3 1 0.5
4 2 1
```

Oszczędne przedstawienie

Od	Do	Opis
0	1 2 3	A→B,A→C,A→D
1	0 3	B→A, B→D
2	4	C→E
3	1 2	D→B,D→C

Błądzenie losowe (zwykły)

```
[ 0.2, 0.2, 0.2, 0.2, 0.2 ]
```

```
[ 0.1     , 0.166667, 0.166667, 0.166667, 0.2 ]
[ 0.058333, 0.086111, 0.086111, 0.086111, 0.116667 ]
[ 0.022801, 0.033218, 0.033218, 0.033218, 0.045602 ]
[ 0.006428, 0.009368, 0.009368, 0.009368, 0.012855 ]
[ 0.001321, 0.001925, 0.001925, 0.001925, 0.002642 ]
[ 0.000198, 0.000288, 0.000288, 0.000288, 0.000396 ]
[ 2.158106e-05, 3.145283e-05, 3.145283e-05, 3.145283e-05, 4.316212e-05 ]
[ 1.716041e-06, 2.501004e-06, 2.501004e-06, 2.501004e-06, 3.432081e-06 ]
[ 9.943498e-08, 1.449192e-07, 1.449192e-07, 1.449192e-07, 1.988700e-07 ]
...
[ 0, 0, 0, 0, 0 ]
```

PageRank with Taxation

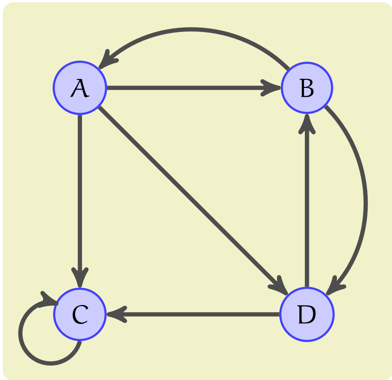
```
[ 0.2, 0.2, 0.2, 0.2, 0.2 ]
[ 0.12, 0.173333, 0.173333, 0.173333, 0.2 ]
[ 0.096533, 0.125689, 0.125689, 0.125689, 0.2 ]
[ 0.084192, 0.107234, 0.107234, 0.107234, 0.2 ]
[ 0.081440, 0.103226, 0.103226, 0.103226, 0.2 ]
[ 0.081105, 0.102738, 0.102738, 0.102738, 0.2 ]
[ 0.081082, 0.102704, 0.102704, 0.102704, 0.2 ]
[ 0.081081, 0.102703, 0.102703, 0.102703, 0.2 ]
[ 0.081081, 0.102703, 0.102703, 0.102703, 0.2 ]
[ 0.081081, 0.102703, 0.102703, 0.102703, 0.2 ]
...
[ 45/555, 57/555, 57/555, 57/555, 111/555 ]
```

Topic-Sensitive PageRank

Zaufany jest węzeł A.

```
[ 0.2, 0.2, 0.2, 0.2, 0.2 ]
[ 0.28    , 0.133333, 0.133333, 0.133333, 0.2 ]
[ 0.2512  , 0.118756, 0.118756, 0.118756, 0.2 ]
[ 0.244718, 0.110264, 0.110264, 0.110264, 0.2 ]
[ 0.243414, 0.108357, 0.108357, 0.108357, 0.2 ]
[ 0.243255, 0.108125, 0.108125, 0.108125, 0.2 ]
[ 0.243244, 0.108109, 0.108109, 0.108109, 0.2 ]
[ 0.243243, 0.108108, 0.108108, 0.108108, 0.2 ]
[ 0.243243, 0.108108, 0.108108, 0.108108, 0.2 ]
[ 0.243243, 0.108108, 0.108108, 0.108108, 0.2 ]
...
[ 45/185, 20/185, 20/185, 20/185, 37/185 ]
```

3.4 Przykład 4



Rysunek 6: Przykład 4.

Macierz sąsiedztwa

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

Macierz przejścia

$$\begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Jest to macierz stochastyczna, ale mamy *spider trap*.

Rzadka macierz przejścia

```
0 1 0.5
1 0 0.33333333
1 3 0.5
2 0 0.33333333
2 2 1
2 3 0.5
3 0 0.33333333
3 1 0.5
```

Oszczędne przedstawienie

Od	Do	Opis
0	1 2 3	A→B,A→C,A→D
1	0 3	B→A, B→D
2	2	C→C
3	1 2	D→B,D→C

Błądzenie losowe (zwykły)

```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.125   , 0.208333, 0.458333, 0.208333 ]
[ 0.072917, 0.107639, 0.711806, 0.107639 ]
[ 0.028501, 0.041522, 0.888455, 0.041522 ]
```

```
[ 0.008035, 0.011710, 0.968546, 0.011710 ]
[ 0.001651, 0.002406, 0.993537, 0.002406 ]
[ 2.472207e-04, 3.603062e-04, 9.990322e-01, 3.603062e-04 ]
[ 2.697633e-05, 3.931603e-05, 9.998944e-01, 3.931603e-05 ]
[ 2.145051e-06, 3.126255e-06, 9.999916e-01, 3.126255e-06 ]
[ 1.242937e-07, 1.811491e-07, 9.999995e-01, 1.811491e-07 ]
..
[ 0, 0, 1, 0 ]
```

PageRank with Taxation

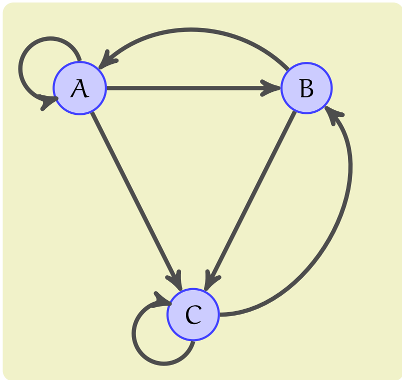
```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.15, 0.216667, 0.416667, 0.216667 ]
[ 0.120667, 0.157111, 0.565111, 0.157111 ]
[ 0.105240, 0.134043, 0.626674, 0.134043 ]
[ 0.101800, 0.129033, 0.640134, 0.129033 ]
[ 0.101382, 0.128422, 0.641774, 0.128422 ]
[ 0.101353, 0.128380, 0.641887, 0.128380 ]
[ 0.101351, 0.128378, 0.641892, 0.128378 ]
[ 0.101351, 0.128378, 0.641892, 0.128378 ]
[ 0.101351, 0.128378, 0.641892, 0.128378 ]
...
[ 15/148, 19/148, 95/148, 19/148 ]
```

Topic-Sensitive PageRank

Zaufany jest węzeł A.

```
[ 0.25, 0.25, 0.25, 0.25 ]
[ 0.3     , 0.166667, 0.366667, 0.166667 ]
[ 0.258667, 0.129778, 0.481778, 0.129778 ]
[ 0.246213, 0.112441, 0.528905, 0.112441 ]
[ 0.243587, 0.108608, 0.539197, 0.108608 ]
[ 0.243266, 0.108142, 0.54045 , 0.108142 ]
[ 0.243244, 0.108109, 0.540537, 0.108109 ]
[ 0.243243, 0.108108, 0.54054 , 0.108108 ]
[ 0.243243, 0.108108, 0.540541, 0.108108 ]
[ 0.243243, 0.108108, 0.540541, 0.108108 ]
...
[ 9/37, 4/37, 20/37, 4/37 ]
```

3.5 Przykład 5



Rysunek 7: Przykład 5.

Macierz sąsiedztwa

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

Macierz przejścia

$$\begin{bmatrix} 1/3 & 1/2 & 0 \\ 1/3 & 0 & 1/2 \\ 1/3 & 1/2 & 1/2 \end{bmatrix}$$

Mamy *spider trap*-y.

Rzadka macierz przejścia

```
0 0 0.33333333
0 1 0.5
1 0 0.33333333
1 2 0.5
2 0 0.33333333
2 1 0.5
2 2 0.5
```

Oszczędne przedstawienie

Od	Do	Opis
0	0 1 2	A→A,A→B,A→C
1	0 2	B→A,B→C
2	1 2	C→B,C→C

Błądzenie losowe (zwykły)

```
[ 0.333333, 0.333333, 0.333333 ]
[ 0.277778, 0.277778, 0.444444 ]
[ 0.234568, 0.304012, 0.461420 ]
[ 0.230653, 0.307877, 0.461470 ]
[ 0.230766, 0.307696, 0.461537 ]
[ 0.230769, 0.307692, 0.461538 ]
[ 0.230769, 0.307692, 0.461538 ]
```

```
[ 0.230769, 0.307692, 0.461538 ]
[ 0.230769, 0.307692, 0.461538 ]
[ 0.230769, 0.307692, 0.461538 ]
...
[ 3/13, 4/13, 6/13 ]
```

PageRank with Taxation

```
[ 0.333333, 0.333333, 0.333333 ]
[ 0.288889, 0.155556, 0.333333 ]
[ 0.179062, 0.121580, 0.333333 ]
[ 0.152302, 0.108243, 0.333333 ]
[ 0.149125, 0.106486, 0.333333 ]
[ 0.148941, 0.106386, 0.333333 ]
[ 0.148936, 0.106383, 0.333333 ]
[ 0.148936, 0.106383, 0.333333 ]
[ 0.148936, 0.106383, 0.333333 ]
[ 0.148936, 0.106383, 0.333333 ]
...
[ 21/141, 15/141, 47/141 ]
```

Topic-Sensitive PageRank

Zaufany jest węzeł B.

```
[ 0.333333, 0.333333, 0.333333 ]
[ 0.222222, 0.288889, 0.333333 ]
[ 0.150321, 0.246617, 0.333333 ]
[ 0.130273, 0.235475, 0.333333 ]
[ 0.127805, 0.234123, 0.333333 ]
[ 0.127664, 0.234045, 0.333333 ]
[ 0.127660, 0.234043, 0.333333 ]
[ 0.127660, 0.234043, 0.333333 ]
[ 0.127660, 0.234043, 0.333333 ]
[ 0.127660, 0.234043, 0.333333 ]
...
[ 18/141, 33/141, 47/141 ]
```