ALGORYTMICZNA I STATYSTYCZNA ANALIZA DANYCH

15/01/2015

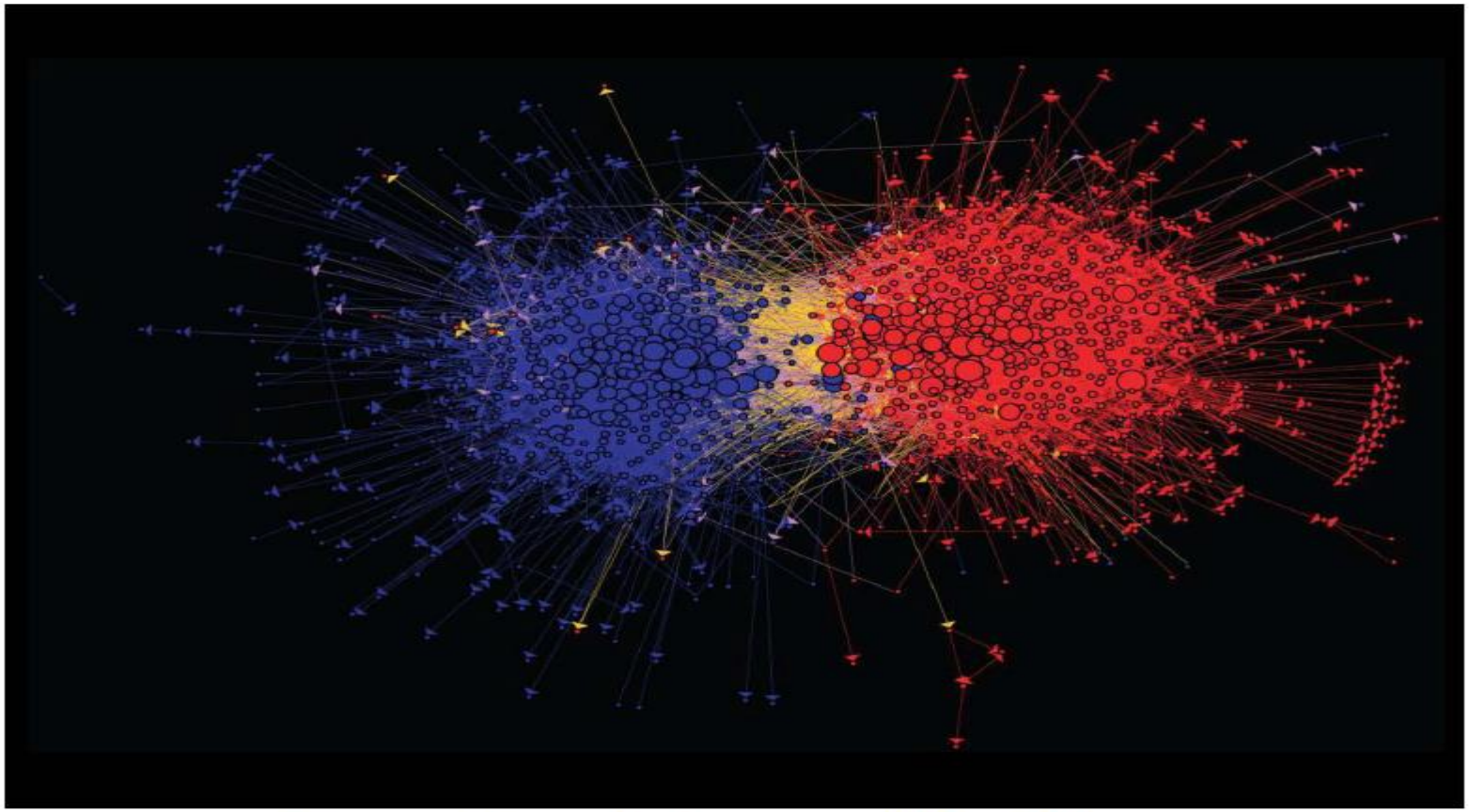
WFAiS UJ, Informatyka Stosowana
II stopień studiów

Dane w postaci grafów

Przykład: social network

Facebook social graph

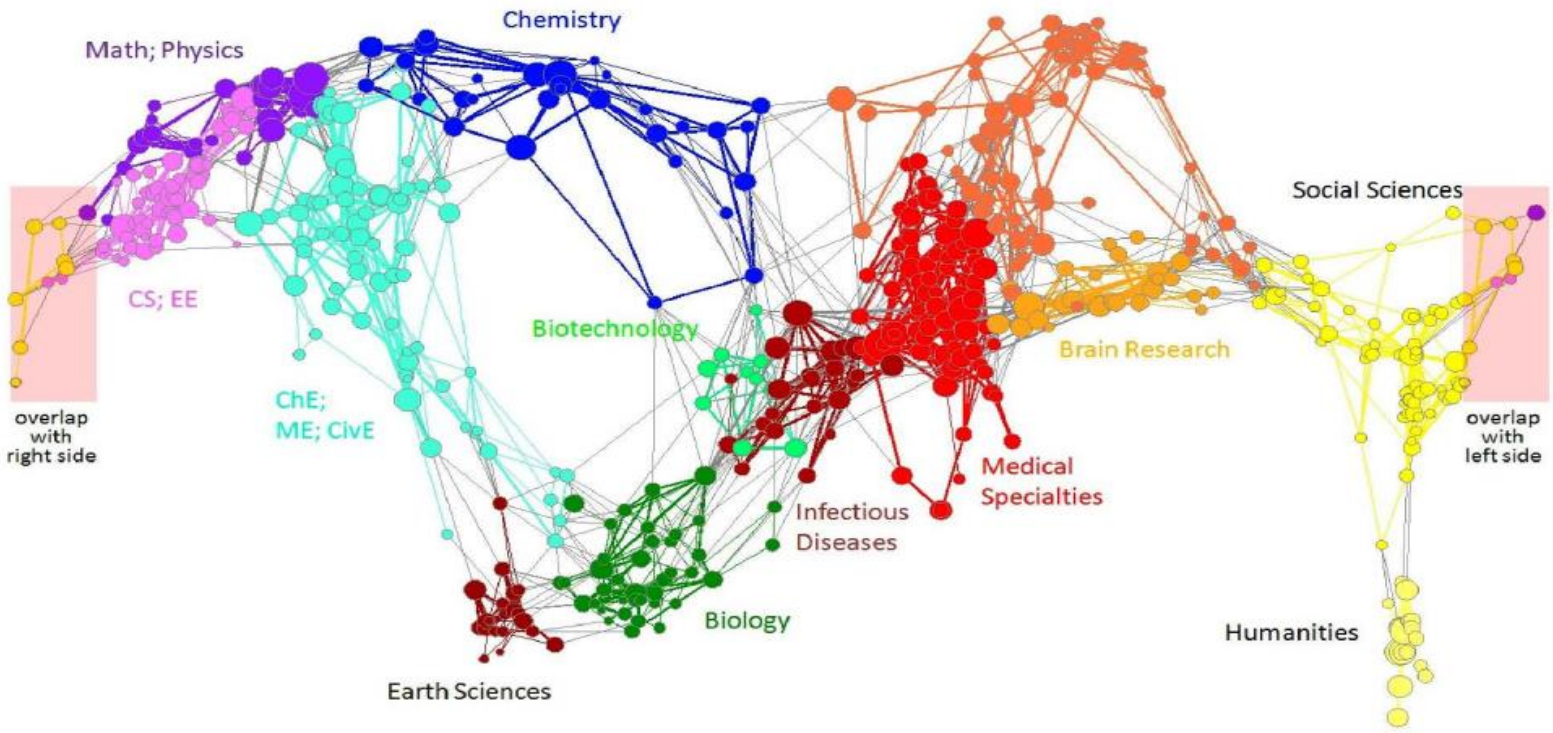
Przykład: media network



Connections between political blogs

Polarization of the network [Adamic-Glance, 2005]

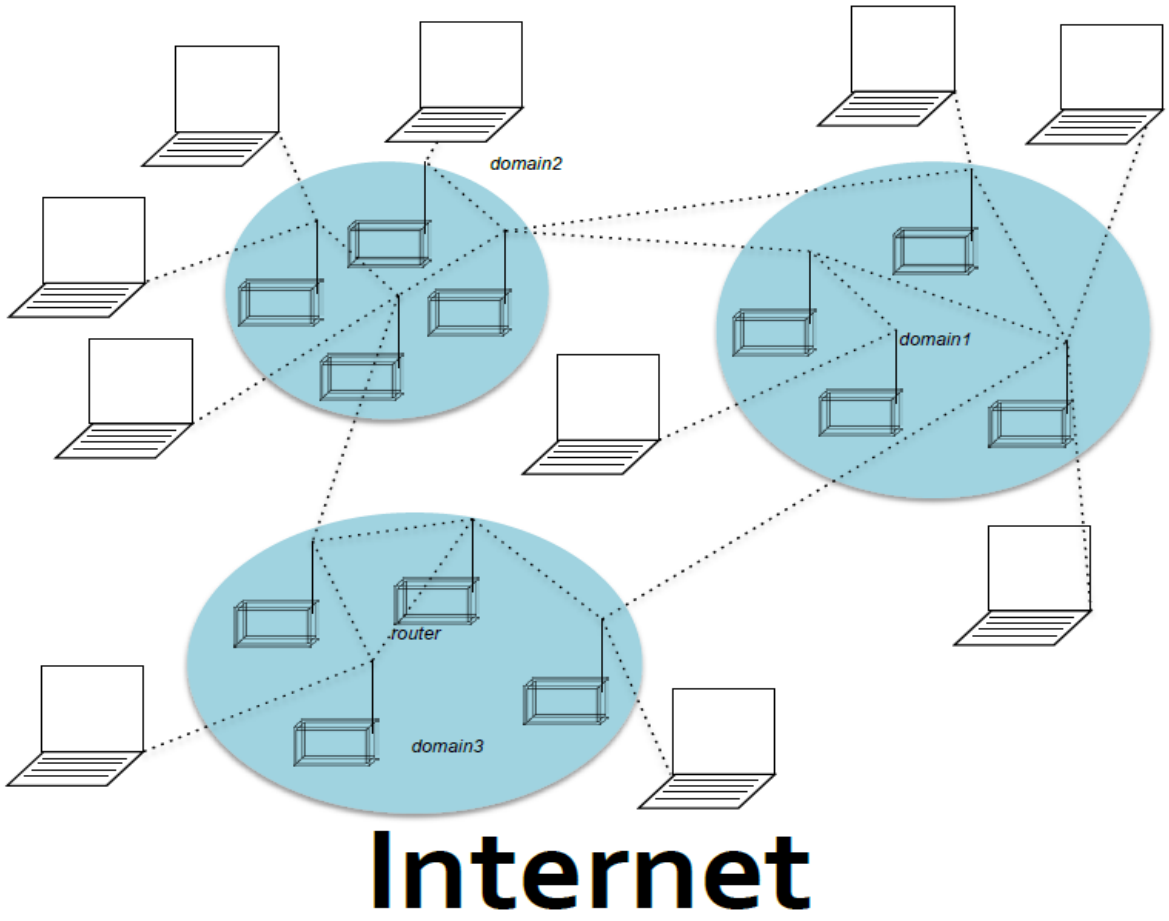
Przykład: information network



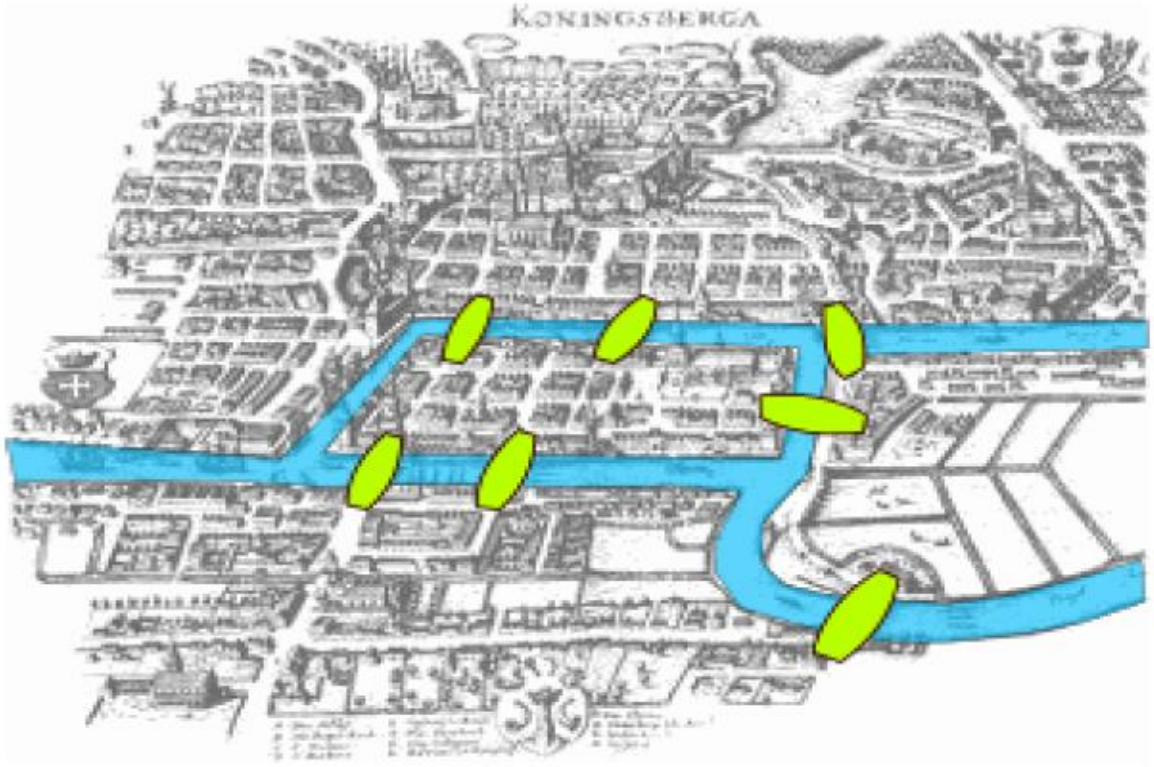
Citation networks and Maps of science

[Börner et al., 2012]

Przykład: communication network



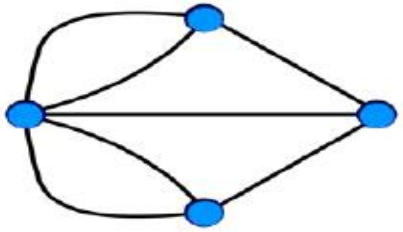
Przykład: technological network



Seven Bridges of Königsberg

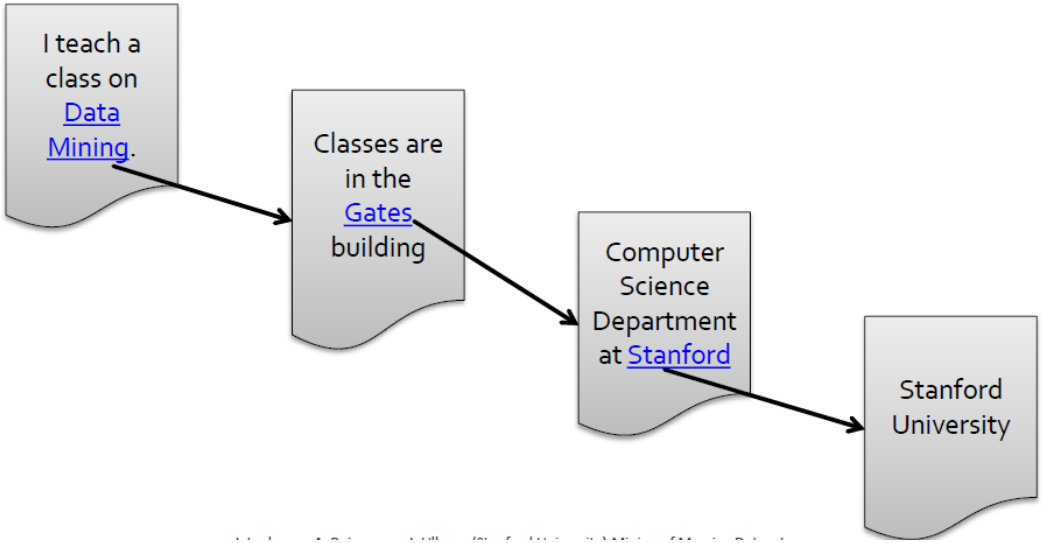
[Euler, 1735]

Return to the starting point by traveling each link of the graph once and only once.

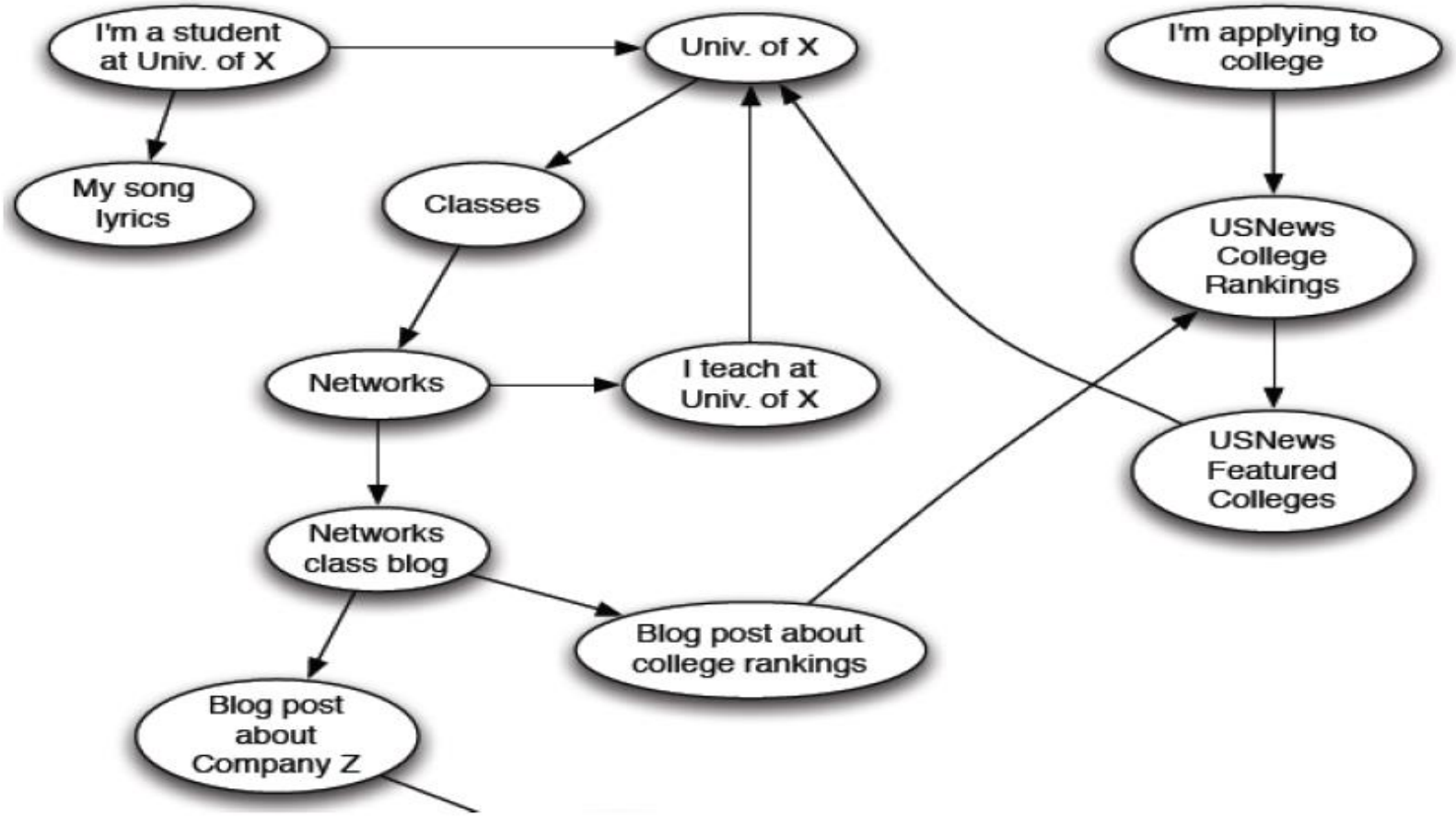


Web jako graf

- Web jako skierowany graf
  - Węzły: strony internetowe
  - Strzałki: hyperlinki



Web jako skierowany graf



Jak organizować web

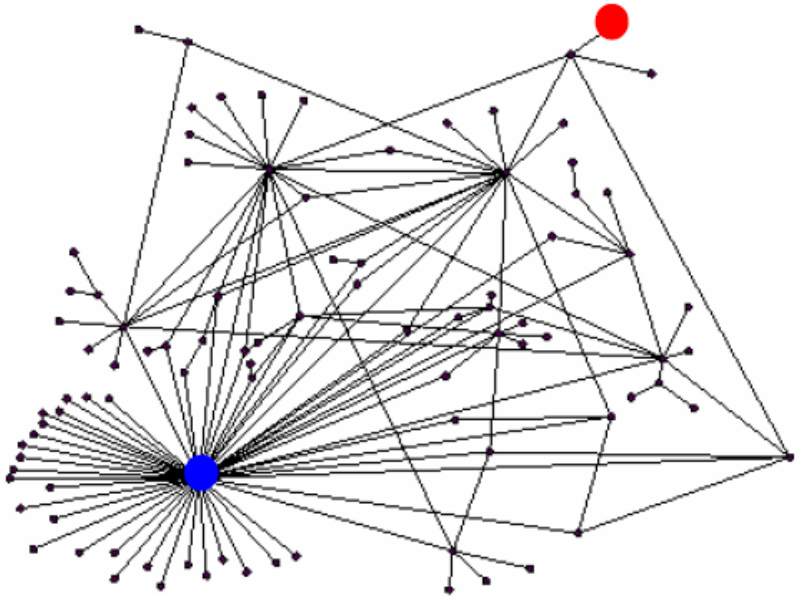
- Pierwsza próba: ręcznie tworzone katalogi
  - Yahoo, DMOZ, LookSmart
- Następna próba: WebSearch
  - Przeszukiwanie zawartości stron z małych i wiarygodnych podzbiorów: artykuły w gazetach, patenty, etc.
  - Ale: Web jest ogromny, wiarygodność stron trudna do weryfikacji, spamowanie web, itd.

Podstawowe wyzwania

- W jaki sposób sprawdzać że strona jest wiarygodna?
- Jaka jest najlepsza odpowiedź na hasło „gazeta”?
  - Nie ma jednej najlepszej odpowiedzi
  - Strony które pokazują informacje na temat gazet mogą dotyczyć wielu różnych gazet

Które strony są najważniejsze?

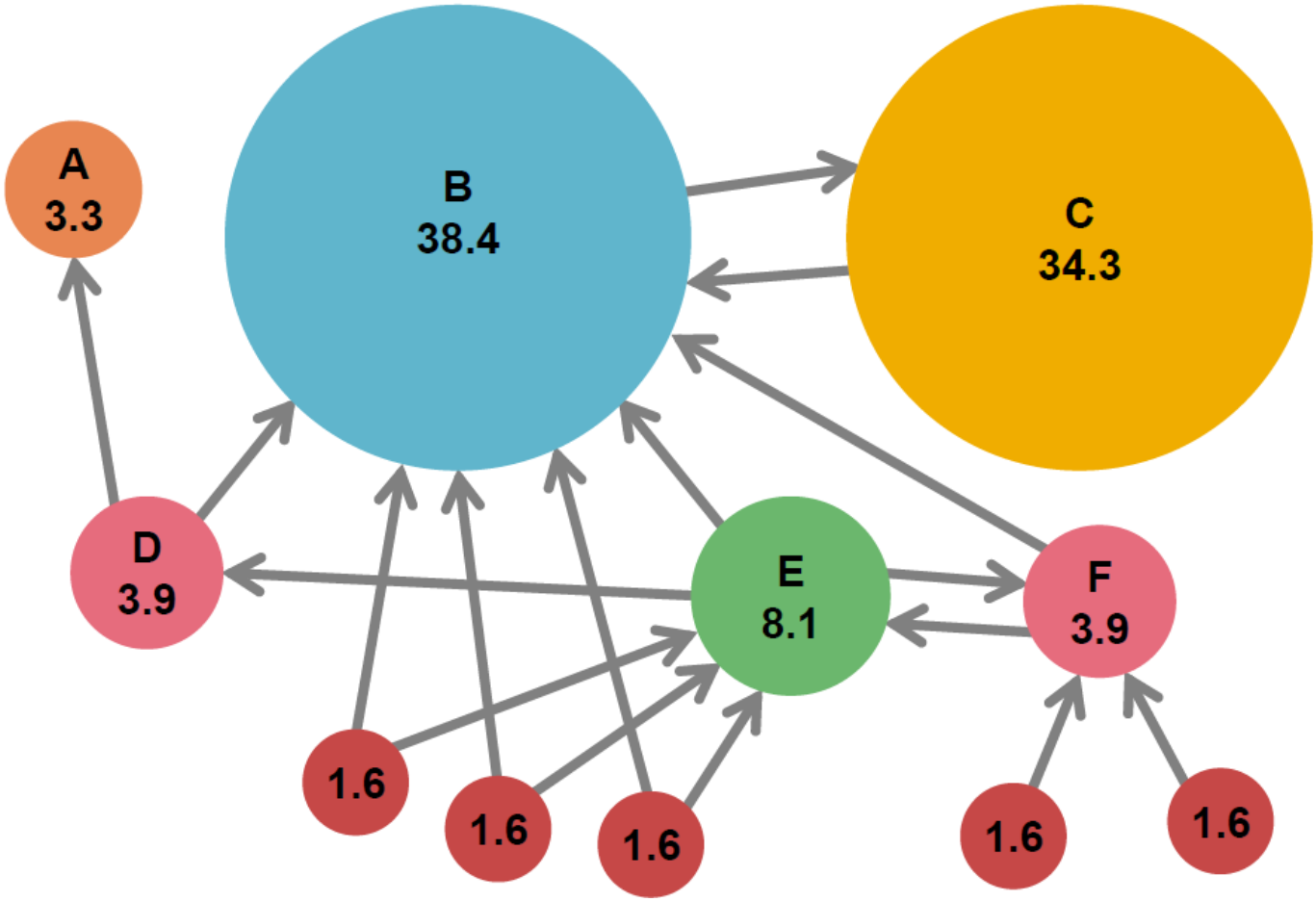
- Nie wszystkie strony są tak samo ważne
- Zróbmy „ranking” stron ze względu na ilość linków



Algorytmy do analizy linków

- Wiele różnych algorytmów zostało rozwiniętych
- Omówię algorytm PageRank używany przez Google.
  - Każdy link liczy się jako punkt, bardziej ważna strona to ta która ma więcej linków: Wchodzących? Wychodzących?
  - A linki a ważnych stron do danej strony powinny się liczyć bardziej! Hm.. To wygląda jak pytanie rekurencyjne..

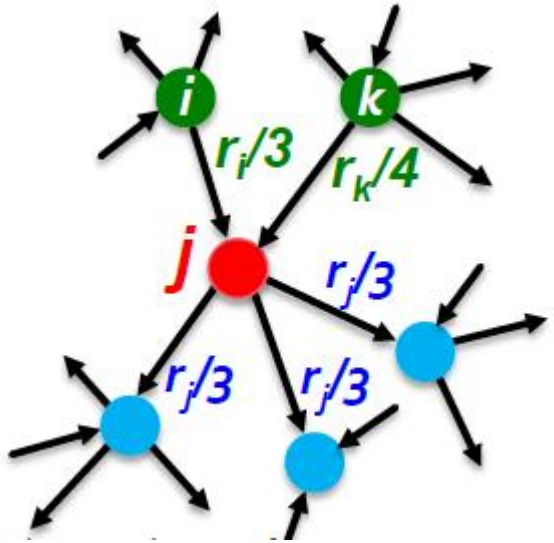
Przykład: PageRank punktacja



Prosta rekurencyjna formuła

- Wartość każdego linku jest proporcjonalna do wartości strony z której wychodzi
- Jeżeli strona **j** z wartością r_j na **n**-wychodzących linków to każdy ma wartość r_j/n.
- Wartość strony **j** jest sumą wartości linków do niej wchodzących.

$$r_j = r_i/3 + r_k/4$$

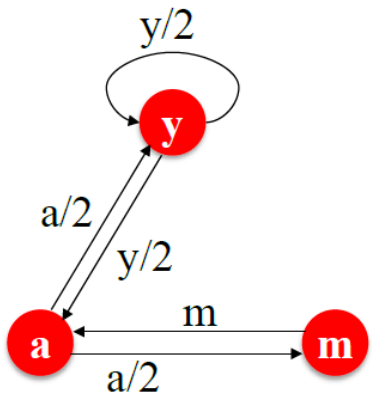


PageRank: flow model

- Link z „ważnej” strony jest więcej warty
- Strona jest „ważna” jeżeli jest wskazywana przez wiele innych stron.
- Zdefiniujemy „rank” strony

$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d_i}}$$

d_i … out-degree of node i



"Flow" equations:

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

Rozwiązanie dla „flow” równania

- 3 równiania, 3 niewiadome, nie ma stałej
  - Nie ma jednego rozwiązania
  - Dodatkowy warunek dot. Normalizacji

Flow equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

  - $r_y + r_a + r_m = 1$
  - Solution: $r_y = \frac{2}{5},\ r_a = \frac{2}{5},\ r_m = \frac{1}{5}$

  - Układ równań możemy rozwiązać np. metodą eliminacji Gaussa. Ale potrzebujemy czegoś lepszego

Interpretacja macierzowa

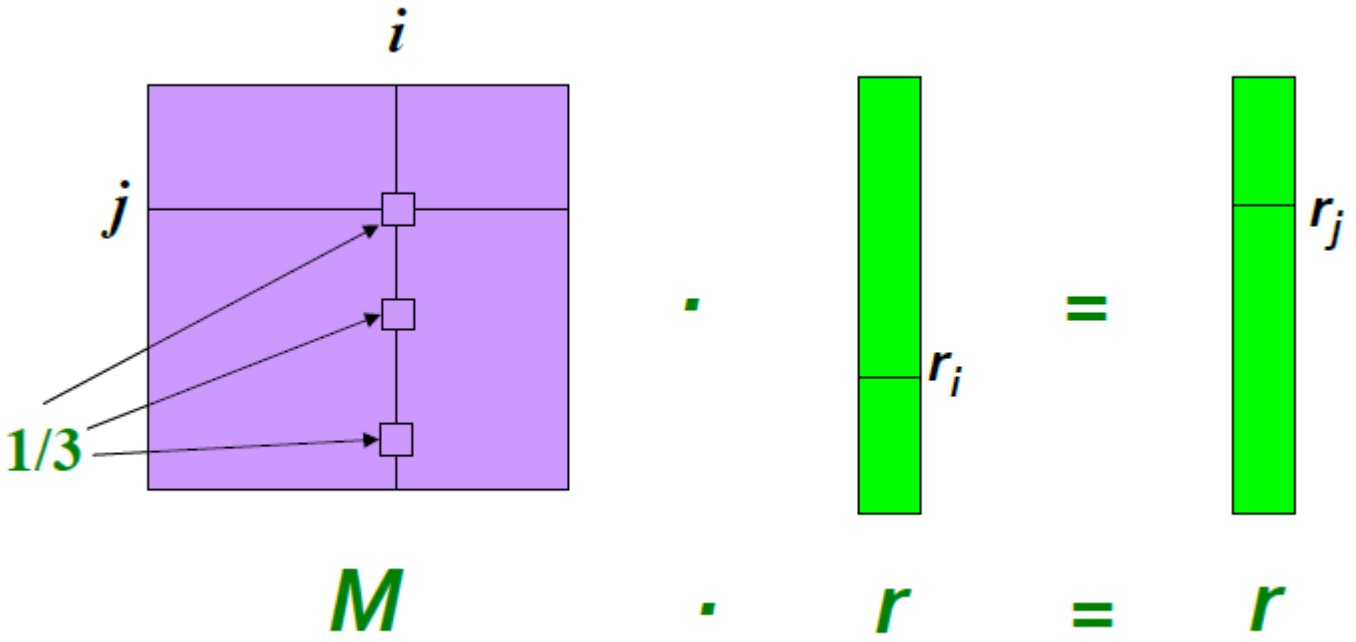
- Ponieważ kolumny normalizowane do 1 => interpretacja prawdopodobieństwowa.
  - Page **i** ma $\mathbf{d_i}$ out-link
  - Jeżeli **i -> j**, to $\mathbf{M_{ji} = 1/d_i}$, w pozostałych $\mathbf{M_{ji} = 0}$
  - Kolumny tej macierzy sumują się do 1
- Oznaczmy r_i = rank strony
- Równanie „flow”

$$\sum_i \boldsymbol{r_i} = \mathbf{1}$$

$$\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$$

$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d_i}}$$

Przykład



Power iteration

- Jak efektywnie możemy rozwiązać takie równanie?
- Metoda „power iteration”

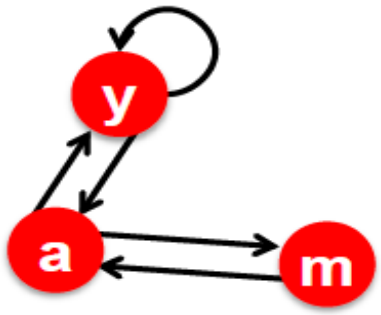
$$\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$$

- Wektor **r** jest wektorem własnym macierzy prawdopodobieństwa.

NOTE: x is an eigenvector with the corresponding eigenvalue λ if:

$$Ax = \lambda x$$

Przykład



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$r = M \cdot r$$

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 1 \\ 0 & ½ & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

Power iteration metoda

- Mając dany graf z N węzłami, gdzie każdy węzeł to są strony a skierowane krawędzie to są hiperlinki
  - Zainicjalizuj $r(0) = [1/N, \ldots. 1/N]^T$
  - Iteruj: $r^{(t+1)} = M\, r^{(t)}$
  - Zatrzymaj jeżeli $| r^{(t+1)} - r^{(t)} |_1 < \varepsilon$

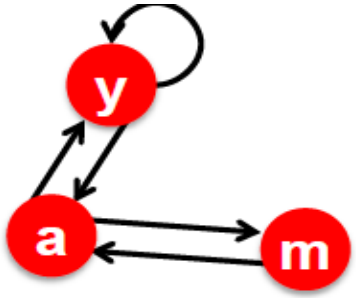
$|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ is the $\mathbf{L_1}$ norm

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

d_i …. out-degree of node i

Przykład

- **Power Iteration:**
  - Set $r_j = 1/N$
  - **1:** $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** $r = r'$
  - If not converged: goto **1**
- **Example:**



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

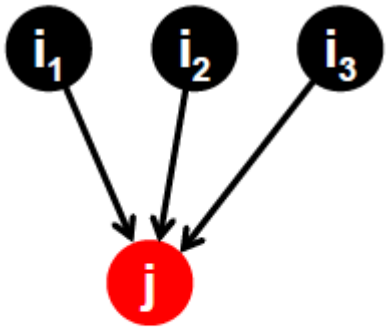
$r_y = r_y/2 + r_a/2$
$r_a = r_y/2 + r_m$
$r_m = r_a/2$

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{matrix}$$

Iteration 0, 1, 2, …

Błądzenie przypadkowe

- Oglądamy przypadkowe strony internetowe
  - W momencie **t**, jesteśmy na stronie **i**
  - W czasie **(t+1)** przechodzimy losowo na jedna ze stron podłączonych ze strony **i**
  - W pewnym momencie kończymy na stronie **j**
  - Powtarzamy ten krok nieskończoną ilość razy.

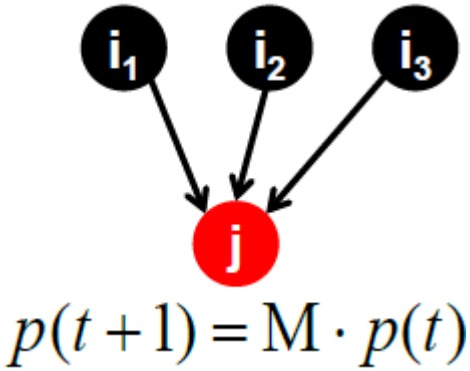


$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_{\mathrm{out}}(\mathrm{i})}$$

Rozkład stacjonarny

- Na jakiej stronie jesteśmy w czasie (t+1)?
  - Błądzimy losowo

$$\boldsymbol{p(t+1) = M \cdot p(t)}$$



  - Załóżmy że rozwiązanie jest stacjonarne

$$\boldsymbol{p(t+1) = M \cdot p(t) = p(t)}$$

  - Nasze orginalne równanie to było

$$\boldsymbol{r = M \cdot r}$$

Procesy Markova

- **Dla grafów które spełniają pewne warunki rozwiązanie stacjonarne będzie osiągnięte niezależnie od warunków początkowych.**
- **To rozwiazanie jest jednoznaczne.**

PageRank wg. Googla

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

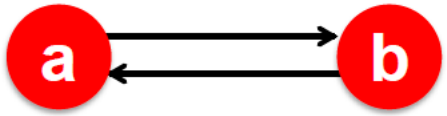
or equivalently

$$r = Mr$$

- Trzy pytania:
  - Czy rozwiązanie zbieżne?
  - Czy zbieżne do rozwiązania które oczekujemy?
  - Czy rozwiązanie jest rozsądne?

Czy rozwiązanie jest zbieżne?

- „Spider trap” problem



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

- **Example:**

r_a	=	1	0	1	0
r_b		0	1	0	1

Iteration 0, 1, 2, …

Czy zbieżne do tego co oczekujemy?

- „Dead end” problem

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

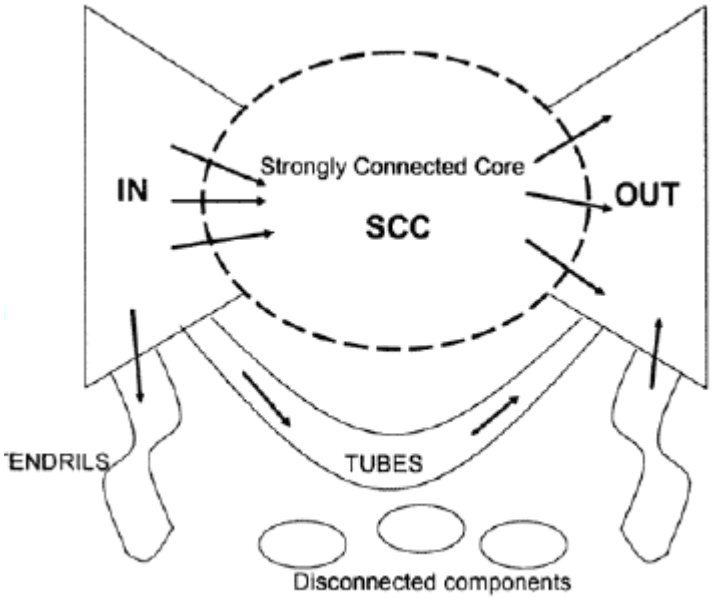
- **Example:**

$$\begin{matrix} r_a \\ r_b \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Iteration 0, 1, 2, …

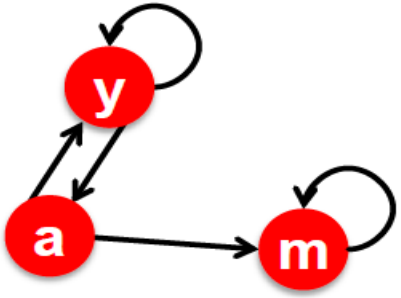
PageRank: problemy

- Niektóre strony są „dead end” czyli nie mają wychodzacych linków
  - Takie strony powodują „wyciekanie” informacji o ważności stron
- Niektóre grupy stron tworzą „spider traps” czyli wszystkie out-linki są do zamkniętej grupy stron.
  - I wtedy tak grupa stron zaasimiluje wszystkie punkty ważności stron.



Spider Traps

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

$r_y = r_y/2 + r_a/2$
$r_a = r_y/2$
$r_m = r_a/2 + r_m$

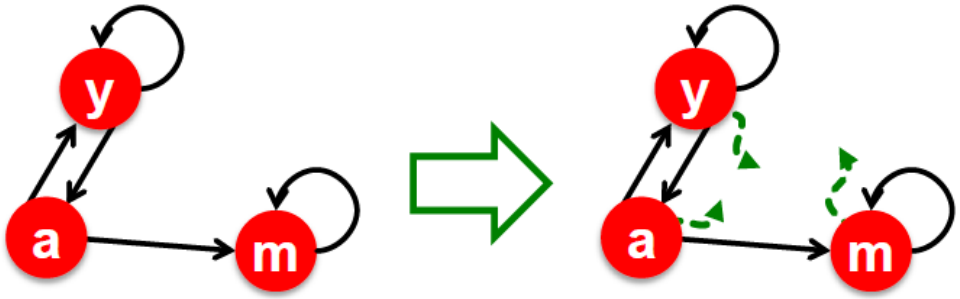
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

Iteration 0, 1, 2, …

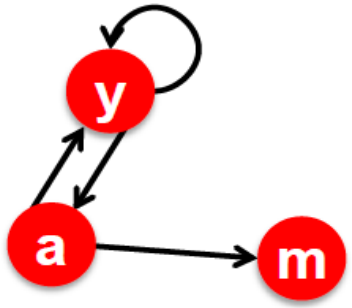
Rozwiązanie: teleportacja

- W każdym kroku błądzenia można
  - z prawdopodobieństwem β pójść jedną z wychodzących ścieżek
  - z prawdopodobieństwem $(1 - \beta)$ przeskoczyć na losowo wybraną stronę
  - Najczęściej stosowane wartości $\beta = 0.8\text{-}0.9$



Problem: „dead end”

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

$r_y = r_y/2 + r_a/2$
$r_a = r_y/2$
$r_m = r_a/2$

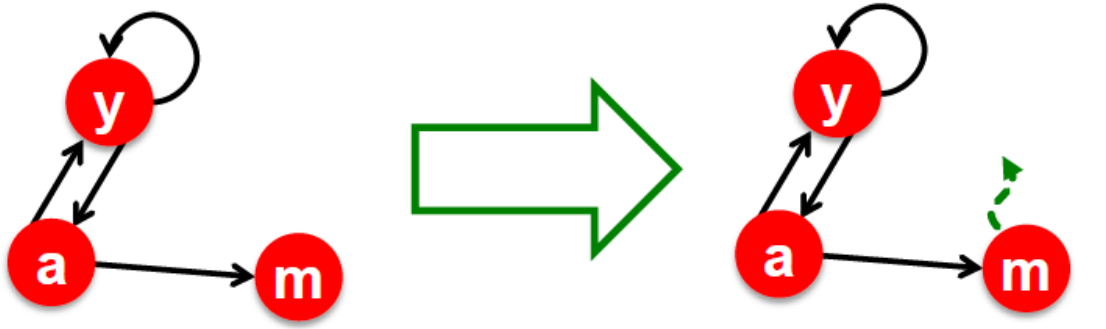
- **Example:**

$$\begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & 0 \end{matrix}$$

Iteration 0, 1, 2, …

Rozwiązanie: teleportacja

- Zawsze przeskocz do losowo wybranej strony
- Odpowiednio zmodyfikuj macierz przejść



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

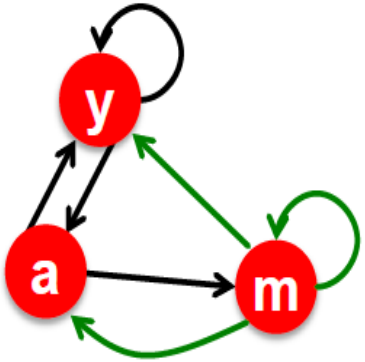
	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

Teleportacja

- Powoduje że macierz staje się stochastyczna

$$A = M + a^T\left(\frac{1}{n}e\right)$$

- a_i…=1 if node *i* has out deg 0, =0 else
- **e**…vector of all 1s



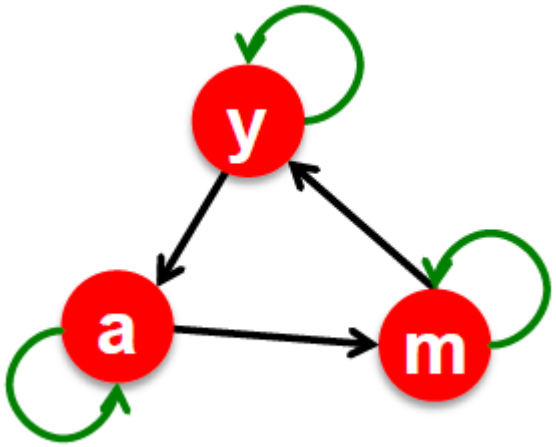
	y	a	m
y	½	½	1/3
a	½	0	1/3
m	0	½	1/3

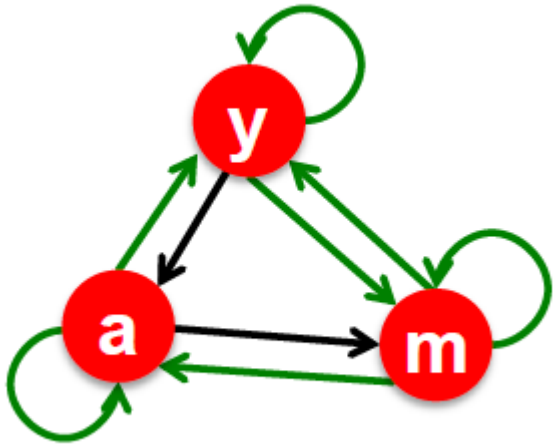
$$r_y = r_y/2 + r_a/2 + r_m/3$$
$$r_a = r_y/2 + r_m/3$$
$$r_m = r_a/2 + r_m/3$$

Teleportacja

- Powoduje że macierz staje się aperiodyczna i nieredukowalna

Niezerowe prawdopodobieństwo Przejścia z każdego stanu do każdego innego.





Rozwiązanie Googla

PageRank equation [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

d_i ... out-degree of node i

The above formulation assumes that M has no dead ends. We can either preprocess matrix M **(bad!)** or explicitly follow random teleport links with probability 1.0 from dead-ends.

Rozwiązanie Googla

- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

- **The Google Matrix *A*:**

$$A = \beta\, M + (1-\beta)\frac{1}{n}\boldsymbol{e} \cdot \boldsymbol{e}^T$$

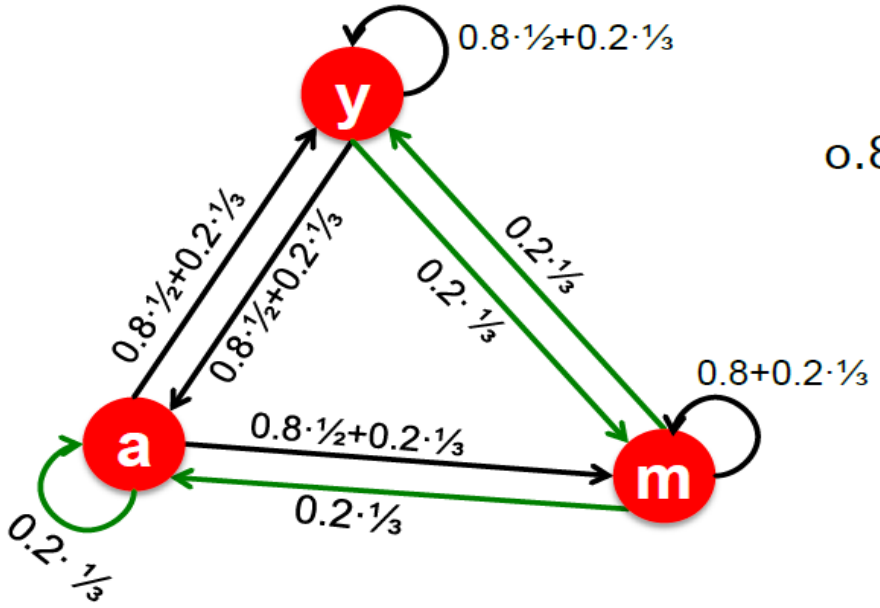
e…vector of all 1s

- ***A* is stochastic, aperiodic and irreducible, so**

$$\boldsymbol{r}^{(t+1)} = \boldsymbol{A} \cdot \boldsymbol{r}^{(t)}$$

- **What is β?**
  - In practice $\beta = 0.8, 0.9$ (make 5 steps and jump)

Przykład



$$M \qquad\qquad 1/n \cdot 1 \cdot 1^T$$

$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{array}{c} \\ y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

$$A$$

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{ccccc} 1/3 & 0.33 & 0.24 & 0.26 & \\ 1/3 & 0.20 & 0.20 & 0.18 & \ldots \\ 1/3 & 0.46 & 0.52 & 0.56 & \end{array} \begin{array}{c} \\ 7/33 \\ 5/33 \\ 21/33 \end{array}$$

PageRank: kompletny algorytm

- **Output: PageRank vector r**
  - **Set:** $r_j^{(0)} = \frac{1}{N}, \quad t = 1$
  - **do:**
    - $\forall j$: $r'^{(t)}_j = \sum_{i \to j} \beta \frac{r_i^{(t-1)}}{d_i}$

      $r'^{(t)}_j = 0$ if in-deg. of j is 0
    - **Now re-insert the leaked PageRank:**

      $\forall j$: $r_j^{(t)} = r'^{(t)}_j + \frac{1-S}{N}$ where: $S = \sum_j r'^{(t)}_j$
    - $t = t + 1$
  - **while** $\sum_j \left| r_j^{(t)} - r_j^{(t-1)} \right| > \varepsilon$