

Data science - laboratorium 2017-10-27

Obliczeniowa nauka o sieciach

Prowadzący: Radosław Michalski

Kontakt: radoslaw.michalski@pwr.edu.pl / <https://www.ii.pwr.edu.pl/~michalski>

Github: <https://github.com/rmhere>

Prace dyplomowe: <https://www.ii.pwr.edu.pl/~michalski/index.php?content=theses>

Cel laboratorium:

- poznanie środowiska R oraz IDE RStudio
- podstawowe typy danych i operacje w środowisku R
- zrozumienie podstaw pakietu igraph dla środowiska R
- tworzenie grafów wedle modelu i z dostępnych źródeł danych
- podstawowe operacje na grafach
- symulacja procesu rozprzestrzeniania się informacji w sieci społecznej

Plan laboratorium:

1. Mini-wykład prowadzącego o języku R, RStudio, podstawowych typach danych i operacjach w R oraz o pakiecie igraph, demonstracja kodu
2. Zadania do samodzielnego wykonania przez studenta
3. Zapisanie kodu i wysłanie mailem do prowadzącego (adres mailowy powyżej)
4. Dyskusja, podsumowanie

Zadania do wykonania:

I Grafy losowe (Erdős-Rényi)

1. Wygeneruj sieć Erdős-Rényi o stu wierzchołkach i prawdopodobieństwie krawędzi = 0.05.
2. Wydrukuj podsumowanie grafu - czy graf jest ważony?
3. Wylistuj wszystkie wierzchołki i krawędzie.
4. Ustaw wagi wszystkich krawędzi na losowe z zakresu 0.01 do 1
5. Wydrukuj ponownie podsumowanie grafu - czy teraz graf jest ważony?
6. Jaki jest stopień każdego węzła? Następnie stwórz histogram stopni węzłów.
7. Ile jest klastrow (connected components) w grafie?
8. Zwizualizuj graf w taki sposób, aby rozmiar węzłów odpowiadał mierze PageRank.

II Grafy preferential attachment (Barabási-Albert)

1. Wygeneruj graf wedle modelu Barabási-Albert z tysiącem węzłów
2. Zwizualizuj graf layoutem Fruchterman & Reingold
3. Znajdź najbardziej centralny węzeł według miary betweenness, jaki ma numer?
4. Jaka jest średnica grafu?
5. W komentarzu napisz czym różni się grafy Barabási-Albert i Erdős-Rényi.

III Rozprzestrzenianie się informacji w sieciach - dane rzeczywiste

1. Pobierz zbiór danych http://konect.uni-koblenz.de/networks/radoslaw_email_email (Downloads, pierwszy od góry, rozpakowany tutaj: https://www.ii.pwr.edu.pl/~michalski/tmp/out.radoslaw_email_email)
2. Zaimportuj zbiór out.radoslaw_email_email do data.frame i zachowaj tylko pierwsze dwie kolumny (dodatkowo przeskocz dwa pierwsze wiersze), następnie stwórz z tego data frame'a graf.

3. Użyj funkcji *simplify* aby pozbyć się wielokrotnych krawędzi i pętli. Zweryfikuj czy po tej operacji Twój graf ma 167 węzłów i 5783 krawędzie. Jeśli tak jest, możesz kontynuować.
4. Zasymuluj proces rozprzestrzeniania się informacji w grafie wedle następującego algorytmu:
 - ustaw wszystkim węzłom atrybut *activated* na FALSE
 - następnie wylosuj jeden węzeł z grafu i ustaw mu atrybut *activated* na TRUE
 - rozpocznij proces rozprzestrzeniania się informacji w grafie (10 przebiegów):
węzeł, który jest aktywowany, aktywuje wszystkich swoich sąsiadów (użyj atrybutu pomocniczego aby w danej iteracji nie aktywować sąsiadów węzła, który został aktywowany w tej samej iteracji)
 - po każdej iteracji zapisz liczbę aktywowanych węzłów w grafie
5. Wykonaj powyższy eksperyment dla pięciu różnych węzłów początkowych, w tym dla najbardziej centralnego węzła wedle metody *betweenness*.
6. Zapisz wynik wszystkich przebiegów w jednym data frame i w pliku, który także dołącz do sprawozdania.

IV Przesłanie plików wynikowych

Prześlij utworzone pliki (skrypt oraz wyniki procesu rozprzestrzeniania się informacji mailem do prowadzącego zajęcia).

Przydatne polecenia:

```
library, read.csv, read.csv2, betweenness, hist, degree, summary, plot,  
erdos.renyi.game, clusters, neighbors, graph.neighborhood,  
graph.data.frame, page.rank, V, E, diameter, simplify, barabasi.game,  
vcount, ecount, sample, rbind
```

Wskazówki:

Pamiętaj, że pomoc do każdego polecenia można uzyskać używając znaku zapytania, np. `?degree`.

Ustawianie atrybutów węzłów:

```
V(g)$atrybut <- wartosc
```

Pętle:

```
E(g)$weight <- runif(length(E(g)), 0.01, 1)  
  
for(i in 1:vcount(g)) {  
  
    print(E(g)[i]$weight)  
}
```

Layouty do wizualizacji:

```
layout.circle, layout.fruchterman.reingold
```

Porównanie (tutaj znajduje te węzły o *betweenness* równym siedem):

```
V(g)[betweenness(g)==7]
```