Hľadanie komunít v sieťach II

Beáta Stehlíková

2-EFM-155 Analýza sociálnych sietí

Fakulta matematiky, fyziky a informatiky, UK v Bratislave, 2019

Hľadanie komunít v sieťach

Prístup I

- Kliky
- Jadrá

Prístup II

- *komunita* bez presnej definície, približne: veľa hrán medzi sebou, málo s inými komunitami
- priebežne počas semestra, v domácej úlohe, v projektoch
- funkcie `leading.eigenvector.community` resp. `cluster_eigenvector` a pod.

Teraz

- zhrnieme si prácu s komunitami
- podrobnejšie sa pozrieme na `cluster_eigenvector`
- vysvetlíme si, čo je *modularita*

Modularita

Definícia

- Pracujeme s neorientovanými neváženými sieťami
- Označenia: $A =$ matica susednosti, $m =$ počet hrán
- Inituitívne: modularita bude *(počet hrán vrámci komunít) mínus (očakávaný počet hrán vrámci komunít)* a budeme ju maximalizovať
- Treba upresniť, čo znamená *očakávaný počet*
- Nech P_{ij} je pravdepodobnosť, že vznikne hrana medzi vrcholmi i a j
- Definujme modularitu ako

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - P_{ij}] \delta(g_i, g_j),$$

kde g_i je komunita, do ktorej patrí vrchol i a $\delta(x, y)$ je 1 pre $x = y$ a inak 0

- Treba určiť P_{ij}

Model Erdősa a Rényiho

- Voľba tohto modelu na porovnanie s reálnou sieťou by znamenalo, že $P_{ij} = p$ (konštanta).
- Prirodzená požiadavka: očakávaný počet hrán sa rovná pozorovanému počtu
- Problém - často výrazne iné rozdelenie stupňa vrcholov

```
data(karate)
g <- karate
hist(degree(karate), 20)
```

Cvičenie: Vygenerujte niekoľko grafov Erdősa a Rényiho a porovnajte rozdelenie stupňa vrcholov s rozdelením pre sieť karate klubu (n rovnaké, p také, aby sa očakávaný počet hrán rovnal pozorovanému počtu)

Lepšia voľba

- Označme $k_i = degree_i$ stupeň vrchola i
- Očakávaný stupeň každého vrchola sa bude rovnať pozorovanému stupňu:

$$\sum_j P_{ij} = k_i,$$

- P_{ij} má tvar

$$P_{ij} = f(k_i)f(k_j)$$

- Dostaneme

$$P_{ij} = \frac{k_i k_j}{2m}$$

Výpočet v R-ku

- Funkcia `modularity`

Dá sa použiť:

- pre výstup z algoritmov hľadania komunít
- pre náš vektor zo zaradením vrcholov (číslo komunity, kam patrí)

```
data(kite)
com <- cluster_walktrap(kite)
modularity(com)

com.vektor <- c(rep(1, 7), rep(2, 3))
modularity(kite, com.vektor)
```

Cvičenie: Zobrazte komunity pre `com.vektor` - nakreslite sieť a vrcholy odlíšte farebne podľa komunity. Je toto rozumné delenie vrcholov do komunít? Ako sa to prejaví na modularite?

Výpočet v R-ku

Iný postup pre vlastné komunity - vytvoríme objekt typu `communities` (a pracujeme s ním ďalej ako s výstupom z napríklad `cluster_walktrap`):

```
com2 <- make_clusters(kite, membership = com.vektor)
class(com2)
```

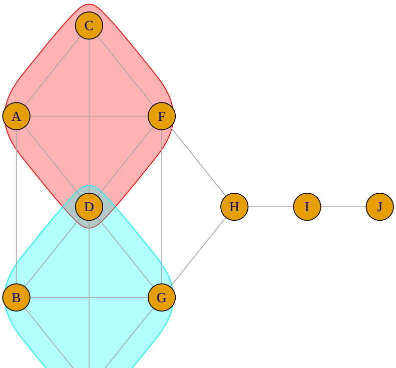
```
## [1] "communities"
```

Potom napr. `plot(com2, kite)`

Vyznačenie skupiny vrcholov v R-ku

Môže byť niekedy užitočné - nevyznačujeme priamo komunity, ale chceme zvýrazniť niektoré vrcholy

```
plot(kite, mark.groups = list(V(kite)[c("A", "C", "D", "F")
                              V(kite)[c("B", "D", "G", "E")
```



Optimalizácia modularity v R-ku

- Metóda `cluster_optimal` priamo optimalizuje modularitu
- V rozumnom čase zbehne len pre menšie siete

Zhlukovanie na základe vlastného vektora

Algoritmus M. E. J. Newmana

M. E. J. Newman: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 036104 (2006)

https://arxiv.org/abs/physics/0605087

Podľa kapitoly 4.A vysvetlite:

- ako súvisia vlastné hodnoty a vektory (a akej matice) s modularitou
- ako z vlastného vektora (a akého) získať rozdelenie vrcholov do dvoch komunít

V R-ku

- funkcia `cluster_leading_eigen` pokračuje ďalších delením (neobmedzuje sa na dve komunity)
- naprogramujeme si delenie do dvoch komunít podľa algoritmu z článku

Cvičenie

Absolútna hodnota vlastného vektora hovorí o tom, "ako silno" patrí vrchol do svojej komunity.

```
set.seed(111)
g <- sample_pa(n=20, directed = FALSE, power=0.5)
g <- add_edges(g, c( 1,5, 17, 5, 6, 4, 6, 11, 6,8, 12, 15,9
lay <- layout.fruchterman.reingold(g)
plot(g, layout=lay, vertex.size=10)
```

- Rozdeľte vrcholy do dvoch komunít podľa vlastného vektora
- Vyznačte v grafe farebne komunity (farbou vrcholov alebo obrysmi pomocou `mark.groups`)
- Veľkosť vrcholov nech je úmerná absolútnej hodnote prvku vlastného vektora (teda veľké vrcholy budú "silne zviazané" so svojou komunitou)

Cvičenie