Network Analysis

Dr. Suat ATAN

07 04 2021

Week 6

- Subgroups & Communities
- Cliques
- K-Core
- Modularity
- Community Detection Methods

Community definition

Wassserman and Faust, Social Network Analysis, Methods and Applications

" A community is subset of actors among whom there are relatively strong, direct, **intense**, frequent of positive ties"

Key Element: The Social Cohesion

- One way to think about network subgroups is through **social cohesion**.
- This approach is so intuitive that it led to a number of the earliest techniques for identifying network subgroups.
- i.e: Cliques

What is a community

- Sometimes visually detectable
- Have more connections among each according to other nodes
- It is not “random”

Where is a community?

The strength of weak ties, Granovetter (1973) suggested that many social networks are made up of relatively densely connected subgroups.

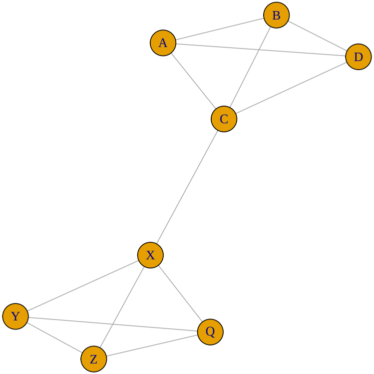
Luke, Douglas A - A User's Guide to Network Analysis

Where is a community?

Many disciplines have theories that assume that larger social systems are made up of distinguishable subgroups, for example sociologists consider social classes; psychologists examine small group behavior, and public health examine health disparities between different social groups.

Luke, Douglas A - A User's Guide to Network Analysis

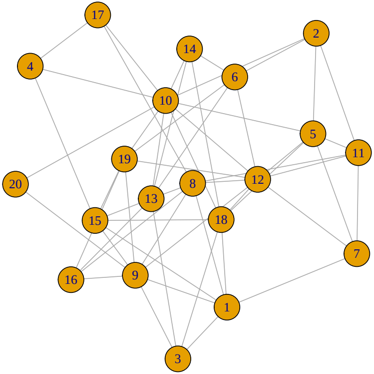
A Community Example



Hidden Communities

- In most real-world social networks the communities are not as visible
- Systematic analysis or observation will be required to reveal it.
- How ?

Hidden Communities (Where are the communities?)



Clique

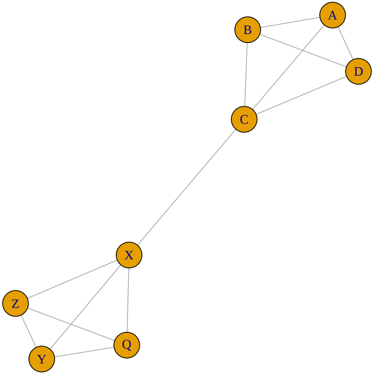
- Cliques are one of the simplest types of cohesive subgroups, and because of their straightforward definition are also one of the easiest types to understand.
- A clique is a **maximally complete subgraph**; that is, it is a subset of nodes that have all possible ties among them.

Luke, Douglas A - A User's Guide to Network Analysis

Clique Example / Tasks:

- Focus on the next slide and think
- Detect the communities visually
- Why they are community?
- What about X and C ?

Clique Example



Cliques

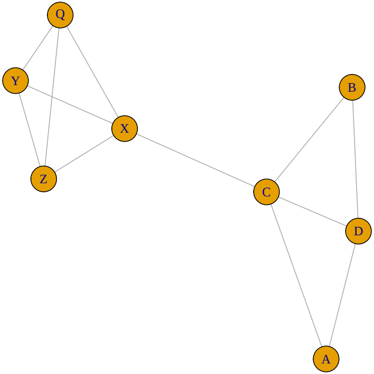
- Consider previous slide. To be a clique, A,B,C,D all of the 6 possible ties must exist between all 4 members.
- If only one is missing, then the seven connections will not belong to one clique.

Clique Definition

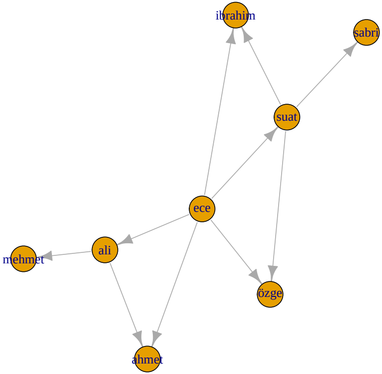
- Fully connected subgroups
- Dyads are fully connected but they are not defined as "clique"
- Isn't very conservative?

Just one missing connection!

It is no longer a clique:



Cliques 1 (R)



Cliques 2 (R)

```
cliques(g2,min = 3)
```

```
## [[1]]
## + 3/8 vertices, named, from 4e9e93b:
## [1] ece      suat     ibrahim
##
## [[2]]
## + 3/8 vertices, named, from 4e9e93b:
## [1] ece   suat özge
##
## [[3]]
## + 3/8 vertices, named, from 4e9e93b:
## [1] ece   ali   ahmet
```

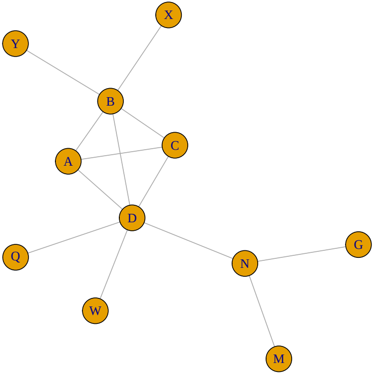
Clique is not realistic definition

Cliques, have major disadvantage that reduce their utility in real-world social network analysis. A clique is a very conservative definition of a community.

K-Core

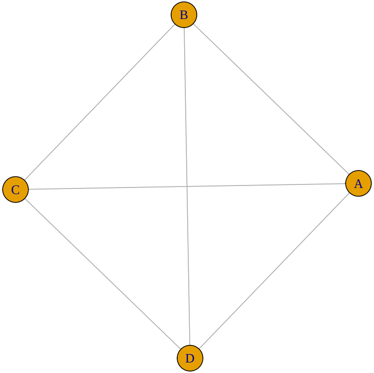
- Variations on the clique concept have been proposed.
- A popular alternative is the k-core.
- **Formal Definition**: A k-core is a **maximal subgraph** where each node is connected to at least k other nodes in the subgraph ?

K-Core (Step by Step Definition)



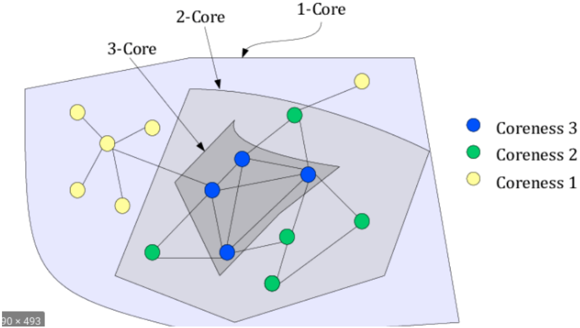
K-Core 1

- Delete the nodes have only one connection
- What remains?
- K-Core: 2 subgraph

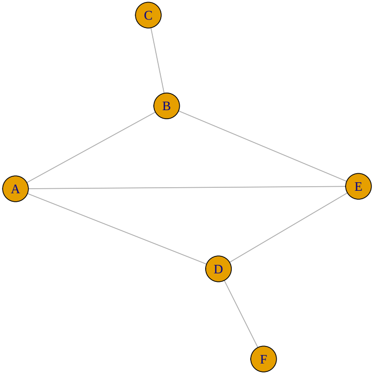


K-Core 2

Distributed k-Core Decomposition, March 2011,IEEE Transactions on Parallel and Distributed Systems 24(2)



K-Core (R)



K-Core (R)

- A,B,E,D are minimum 2 connection. They are members of k-core:2
- C,F is minimum (just) one connection. They are members of k-core:1

```
coreness(golem)
```

```
## A B C D E F
## 2 2 1 2 2 1
```

Colorize K-Core 1

```
golem <- set_vertex_attr(golem,"core",
                         index= V(golem),
                         coreness(golem))
```

Colorize K-Core 2

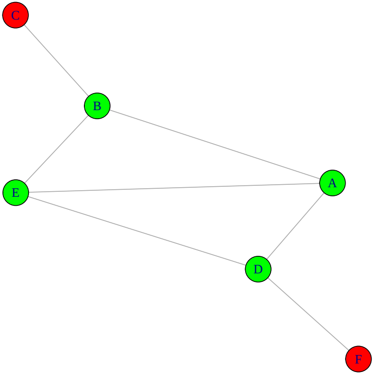
```
## [1] 2 2 1 2 2 1
```

Colorize K-Core 3

```
V(golem)$color = ifelse(V(golem)$core ==1,
                        "red",
                        "green")
V(golem)$color
```

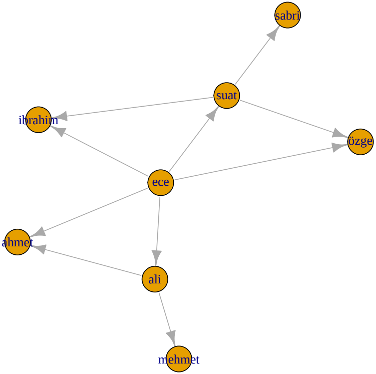
```
## [1] "green" "green" "red"   "green" "green" "red"
```

Colorize K-Core 4



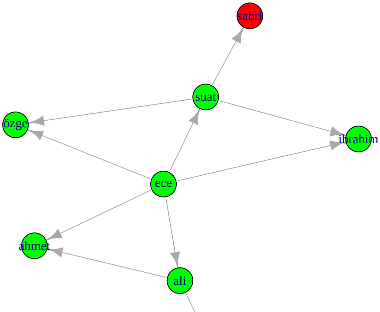
K-Core For Complex Network (R) 1

```
df2 <- read.csv("../data/02.csv")
g2 <- graph_from_data_frame(df2)
plot(g2)
```



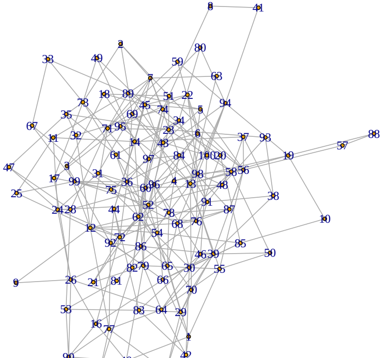
K-Core For Complex Network (R) 2

```
g2 <- set_vertex_attr(g2,"core",
                          index= V(g2),
                          coreness(g2))
V(g2)$color = ifelse(V(g2)$core ==1,"red","green")
plot(g2)
```



K-Core For Random Game (R) 3

```r
bignet <- random.graph.game(n = 100,
                            p.or.m = 1/20,
                            directed = F)
plot(bignet,vertex.label.cex=0.9,vertex.size=2)
```



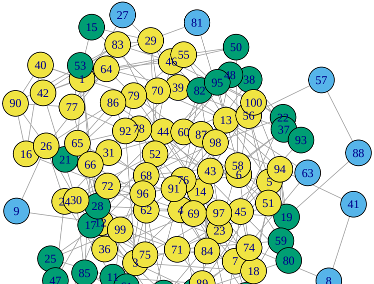
K-Core For Random Game(R) 4

```
coreness(bignet)
```

```
##    [1] 4 3 4 4 4 4 4 2 2 2 3 4 4 4 3 4 3 4 3 3 3 3
##   [38] 3 4 4 2 4 4 4 4 4 3 3 3 3 4 4 3 3 4 4 2 4 3
##   [75] 4 4 4 4 4 3 2 3 4 4 3 4 4 2 4 4 4 4 3 4 3 4
```

K-Core For Random Game(R) 5

```
bignet <- set_vertex_attr(bignet,"core",
                          index= V(bignet),
                          coreness(bignet))
V(bignet)$color = V(bignet)$core
plot(bignet,vertex.label.cex=0.9,
     layout=layout.kamada.kawai(bignet))
```



Modularity Definition

Modularity is a measure of the structure of the net-work, specifically the extent to which nodes exhibit clustering where there is greater density within the clusters and less density between them (Newman2006)

Case 1: Xenophobic

```
moda <- read.csv("../data/modular-a.csv")
moda
```

```
##        person1    person2
## 1  raskolnikov smerdyakov
## 2   smerdyakov     petrov
## 3       petrov   zamyotov
## 4  raskolnikov   zamyotov
## 5  raskolnikov     petrov
## 6       nilgun       ayse
## 7       nilgun        can
## 8          can       ayse
## 9          can       arzu
## 10        arzu     nilgun
## 11         can   zamyotov
```

Case 1: Set origins: Russian or Turkish

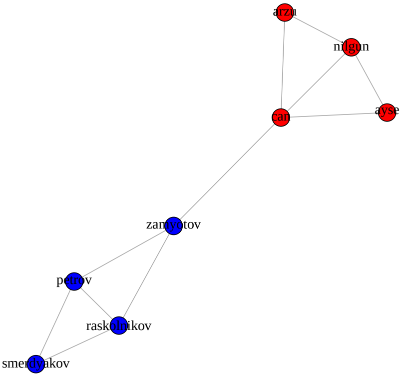
```
gmoda <- graph_from_data_frame(moda,directed = FALSE)
grepl("ov","raskov")
```

```
## [1] TRUE
```

```
V(gmoda)$origin = ifelse(grepl("ov",
                              V(gmoda)$name),"russia
V(gmoda)$color = ifelse(grepl("ov",
                             V(gmoda)$name),"blue","
V(gmoda)$origin_id = ifelse(grepl("ov",
                                 V(gmoda)$name),1,2)
V(gmoda)$origin
```

```
## [1] "russian" "russian" "russian" "turkish" "turki
## [8] "turkish"
```

Case 1


arzu
nilgun
ayse
can
zamyotov
petrov
raskolnikov
smerdyakov

Modularity of Case 1

Calculation with R:

```
modularity(gmoda,V(gmoda)$origin_id,)
```

```
## [1] 0.4090909
```

Parameter 1: Graph object, Parameter 2: Vertices

Case 2: Multicultural

```r
modb <- read.csv("../data/modular-b.csv")
modb
```

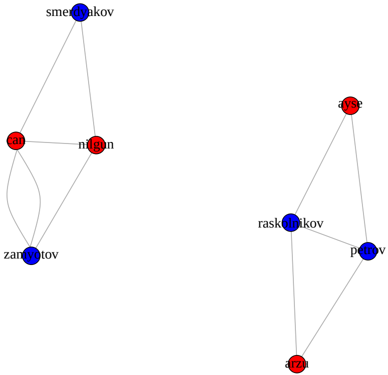
```
##          person1    person2
## 1  raskolnikov       ayse
## 2         ayse     petrov
## 3       petrov       arzu
## 4  raskolnikov       arzu
## 5  raskolnikov     petrov
## 6       nilgun smerdyakov
## 7       nilgun        can
## 8          can smerdyakov
## 9          can   zamyotov
## 10    zamyotov     nilgun
## 11         can   zamyotov
```

Case 2: Set origins

```
gmodb <- graph_from_data_frame(modb,directed = FALSE)
V(gmodb)$origin = ifelse(grepl("ov",
                               V(gmodb)$name),
                         "russian","turkish")
V(gmodb)$color = ifelse(grepl("ov",
                              V(gmodb)$name),
                        "blue","red")
V(gmodb)$origin_id = ifelse(grepl("ov",
                                  V(gmodb)$name),
                            1,2)
V(gmodb)$origin
```

```
## [1] "russian" "turkish" "russian" "turkish" "turki
## [8] "russian"
```

Case 2: Multicultural



Modularity of Case 2

Calculation with R

```
modularity(gmodb,V(gmodb)$origin_id)
```

```
## [1] -0.3181818
```

Parameter 1: Graph object, Parameter 2: Vertices

The End

Thanks

Dr. Suat ATAN