

CAIM Laboratory

Session 7: Introduction to igraph

2019-2020 Q1

Pol Renau Larrodé
Roger Vilaseca Darne

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



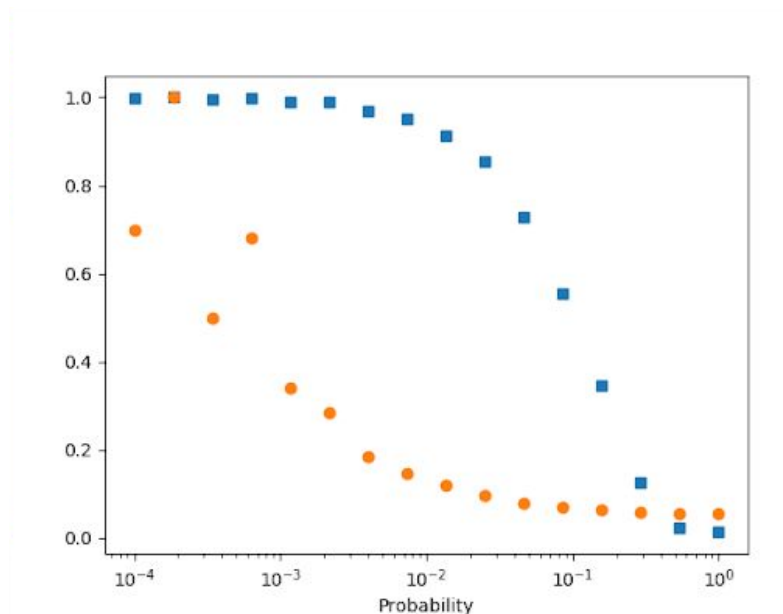
1. Implementació i problemes

- Tota la implementació de la pràctica ha estat realitzada amb python.
- La implementació de la tasca 2, hem realitzat un ús bàsic de la llibreria `igraph`, que ens permet realitzar totes les interaccions necessàries per a la realització de la pràctica. La part de la community detection l'hem realitzat amb el algorisme `community_walktrap`, que té un paràmetre anomenat `steps`, que el que ens indica és el nombre de pasos random que podem realitzar al fer la construcció de la community. Com explicarem més endavant jugarem amb aquest valor per veure els efectes propis sobre el resultat.
- No hem trobat cap problema remarcable en aquesta pràctica, simplement veure com fer el ús adient de la llibreria `igraph`, i realitzar els plots pertinents.

2. Tasques

2.1 Tasca 1

- Per realitzar aquest apartat de la pràctica hem usat la funció `Watts_Strogatz()` de la llibreria `igraph`, primerament ho hem realitzat amb el valor que surt indicat a la documentació de la pràctica (100), no obstant els resultats no eren molt clars així que hem decidit usar un valor superior, 1000. Hem pogut veure que a mesura que augmentem el valor, ens dona una gràfica més similar a la que hem d'obtenir, no obstant el cost temporal és més elevat, per aquest motiu, hem decidit deixar-lo fixat a 1000. Veiem que en l'eix de les y tenim un valor normalitzat, i a la de l'eix de les x una escala logarítmica.



2.2 Tasca 2

- En aquest apartat realitzem un anàlisis del graf donat i la sortida és la següent:

- * Nombre d'arestes: 602
- * Nombre de vèrtexs: 62
- * Diàmetre: 3
- * Transitivitat: 0.5227690047741461
- * Llista de graus:

[63, 13, 9, 39, 35, 7, 9, 59, 13, 25, 13, 11, 33, 31, 19, 21, 27, 7, 23, 33, 7, 9, 13, 35, 37, 17, 13, 13, 17, 9, 35, 23, 25, 33, 31, 3, 21, 11, 11, 13, 21, 11, 23, 29, 43, 15, 15, 23, 15, 7, 15, 15, 7, 19, 7, 15, 7, 9, 15, 17, 3, 7]

- * Graus de llibertat: (posem un petit extracte ja que és bastant extens)

N = 62, mean +- sd: 19.4194 +- 12.5300

[3, 4): ** (2)

[4, 5): (0)

...

[7, 8): ***** (8)

[8, 9): (0)

[9, 10): ***** (5)

[10, 11): (0)

[11, 12): **** (4)

...

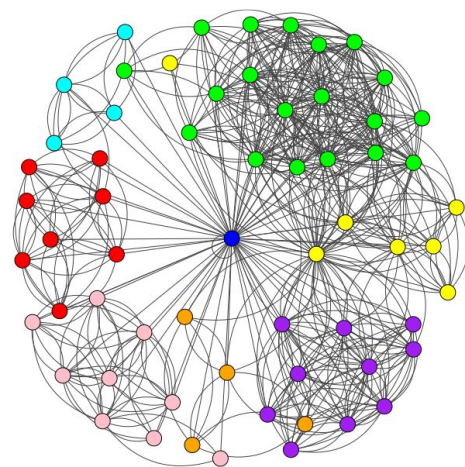
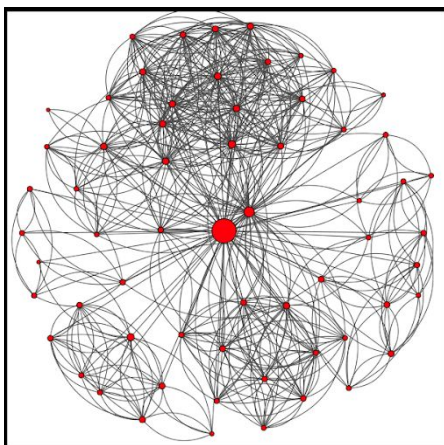
[54, 55): (0)

[55, 56): (0)

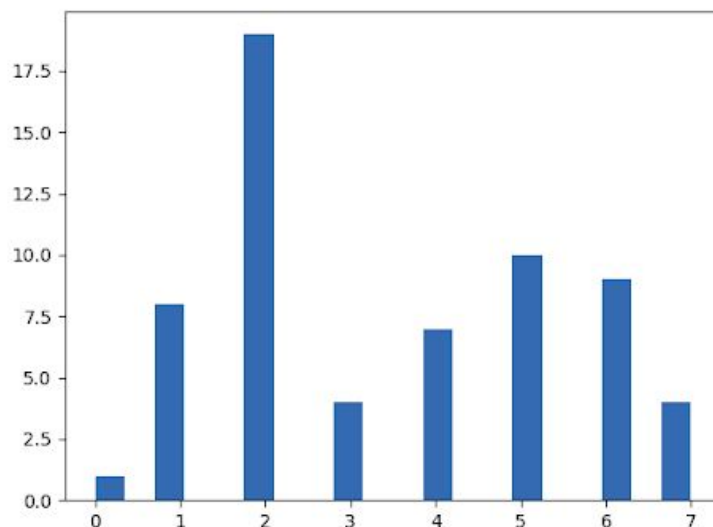
[62, 63): (0)

[63, 64): * (1)

- Per analitzar exhaustivament si el graf és aleatori, printem el graf de manera que el tamany del node és proporcional al pageRank d'aquest.



- Com podem apreciar el node té una distribució poc comú en els grafs aleatoris, ja que tots els nodes tenen aproximadament el mateix valor de pageRank, no obstant hi ha un node central al que apunten tots els nodes, que ens fa pensar que no es una distribució aleatoria. També podem apreciar que podem veure a simple vista algunes comunitats definides, ja que hi ha una densitat d'arestes molt alta en algunes parts del graf, que no apareixen en una distribució aleatoria.
- Podem apreciar que les comunitats que veiem clarament en el graf de pageRank, són les comunitats que el community_walker, ens ha deixat com ha resultat, amb alguna possible variació degut a que té un paràmetre que ens dona la randomicitat d'aquest, i l'hem establert a 4 per voluntat pròpia.
- Com veiem a la imatge anterior, el algorisme de detectar comunitats ens ha deixat un total de 8 comunitats diferents. Analitzarem aquestes comunitats mitjançant histogrames del tamanys d'elles.
- Com veiem al histograma la primera community la de color blau, només té un únic individu, seguidament tenim dues que tenen un total de 4 nodes (la 3 i la 7). La següent comunitat es la que té 7 individus (4), la següent és una que conté 8 individus (1). La comunitat 6 conté 9 nodes, la comunitat 5 té 10, i finalment tenim la classe de major tamany que conté 18 nodes (2).



3. Experimentació

- Hem pogut observar realitzant variacions del paràmetre “número de vèrtex”, en la funció Watts_Strogatz(), cada cop teniem un gràfic més acurat al que havíem d’obtenir, òbviament el cost temporal anava augmentant conforme incrementàvem el nombre de vèrtexs.
- Hem variat el paràmetre steps de la segona part de la tasca, amb valors de 1 a 10. La resta d’elements els hem deixat estàtics.
- Hem apreciat que si variem molt el valor del step, les comunitats que eren clares, no ho són tant degut l’efecte de randomicitat, no obstant si posem valors baixos, les comunitats generades són molt fixades conforme el que veiem a simple vista amb el graf del pageRank, això es degut a que si podem fer menys steps aleatoris els conjunts generats seran aquells que considerem com a reals veien’t-los a simple vista.

4. Conclusions

- Quan tenim una xarxa que té una centralitat bastant distribuïda de forma equitativa (el cas que em vist en aquesta pràctica), hauriem de sospitar que no és una xarxa aleatòria. També veiem que si posem un valor bastant alt en la randomicitat del community, no ens quedaran classes com les desitjades habitualment, no obstant tot depèn de l’objectiu de la segmentació.