Arnau Camprubí
Daniel Navarro

1a part: adquisició i emmagatzematge de les dades

Indiqueu quin node heu seleccionat com a llavor del crawler i per què l'heu seleccionat.

Com a llavor, s'ha seleccionat un node que és un amic de la infància d'un dels participants del projecte. S'escull perquè és bastant aficionat als e-sports i competeix amb un equip de competitiu a un videojoc anomenat Call of Duty: Modern Warfare. Al competir amb un equip, conèixer jugadors de tot Espanya i haver competit en diferents llocs del país, es pensa que pot tenir moltes relacions que no només es mouen per Catalunya, així que em sembla bastant interessant.

Doneu les dades més rellevants del node (nombre de seguidors, nombre d'usuaris als que segueix) així com la url del seu usuari de Twitter.

Nom node	Nombre de seguidors	Nombre d'usuaris als que segueix	Url del seu usuari
RADIIAANT	1332	815	https://twitter.com/RADIIAANT

Expliqueu quines opcions d'exploració heu utilitzat per implementar el vostre crawler. En particular indiqueu quin algorisme d'exploració heu fet servir. Indiqueu també detalls com ara com heu seguit els nodes en cada direcció, si heu fet servir algun ordre especial a l'hora de seleccionar els nodes per a la seva exploració i, en general, les decisions que creieu que són importants de la vostra implementació.

Amb l'objectiu de no allunyar-se molt del node central, es va pensar que explorant dues capes a partir del node central seria suficient com per obtenir informació útil. Així doncs, el meu algorisme es divideix en dues capes:

Per a la primera capa d'exploració, a partir d'un node llavor, l' algorisme recorre els seguidors d'aquest node llavor, quedant-se amb els nodes que tenen més seguidors. L'algorisme en sí, recorre els 20 seguidors més recents d'un node, quedant-se amb els que més seguidors tenen.

La quantitat de nodes amb els que ens quedem la determina el nombre de nodes que volem explorar en total, és a dir, que la quantitat de nodes amb els que ens quedem tant en aquesta primera capa d'exploració com en la segona, és determinada pel paràmetre *max_nodes_to_crawl*. Com més alt sigui el valor d'aquest paràmetre, més nodes seran explorats en aquesta primera capa, i conseqüentment, més explorarem en la segona capa, i a la inversa.

Un cop feta la primera capa d'exploració, amb cadascun d'aquests nodes explorats anteriorment, es realitza exactament el mateix procés que hem fet abans: explorem els 20 seguidors més recents de cadascun d'aquests nodes i ens quedem **només amb el node que més seguidors tingui.** Un cop finalitzat aquest pas, ja haurem explorat dues capes a partir del node central, així que ja haurem acabat.

Com a excepció, pot sortir el dubte de què passa en el cas d'haver d'explorar més de 20 nodes en la primera capa d'exploració. Aquest dubte es pot argumentar dient que si l'algorisme només agafa els 20 seguidors més recents del node llavor i es queda amb els que tenen més seguidors, què passarà si en la primera capa d'exploració necessito explorar 23 nodes? Hi haurà 3 nodes que no es podran explorar? En aquests casos, els 3 nodes que es necessiten de més, seran els 3 nodes (usuaris que el node llavor segueix) amb més seguidors. És a dir, en comptes de seleccionar usuaris de "seguidores", es passen a seleccionar usuaris de "siguiendo".

Expliqueu per què amb un valor relativament petit de nodes explorats acabeu obtenint un graf de mida força gran.

El motiu per el qual amb un valor relativament petit de nodes explorats acabem obtenint un graf de mida (nombre d'arestes) força gran és perquè, tot i que no es el nostre cas, quan s'explora un node es visiten X nodes mes i la presentació demana que n'explorem 40 i en visitem X per cada un d'aquest 40; tenint en compte que al graf es representen tant explorats com visitats el valor d'aquesta X indica el creixement (no lineal) del tamany del graf.

A més gran la X més gran el número de nodes.

Analitzeu els valors màxims del grau d'entrada i del grau de sortida dels nodes que hi ha al graf del fitxer id_seed_node_n.pickle. Veieu algun fet estrany si el compareu amb les dades reals dels usuaris corresponents que es mostren a Twitter? Podeu explicar-lo?

El grau d'entrada d'un node són els usuaris "seguidores" que té, mentre que el grau de sortida són els usuaris "siguiendo" que segueix.

És lògic que si s'analitza els valors màxims del grau d'entrada i del grau de sortida dels nodes del graf que s'han generat amb l'algorisme, obtingui:

- Un màxim en el valor del grau d'entrada de 2 (node anomenat '_PzaiD'), ja que existeix un cas en que un node del primer nivell té el node llavor com a seguidor amb més seguidors. És un cas especial que pot passar, i que si s'agafa el node llavor que s'ha escollit, passa.
- Un màxim en el valor del grau de sortida de 20 (node llavor anomenat 'RADIIAANT'), ja que els màxims nodes a explorar són 40 (paràmetre ja donat de la funció crawler), i el meu algorisme, a l'utilitzar dues capes d'exploració, fa sempre una relació per a saber el nombre de nodes que ha de tenir la primera capa d'exploració. Aquest nombre és el màxim valor del grau de sortida d'un node, que en aquest cas, és el node llavor.

És normal que comparat amb les dades reals dels usuaris corresponents a Twitter aquests valors siguin molt diferents, perquè només s'està estudiant una part dels usuaris, no exs'exploren els usuaris al complet per a saber els "seguidores" i els "seguidos" que té cada usuari.

2a part: preprocessat i visualització de la informació

Per als tres grafs obtinguts:

- **id_seed_node_n.pickle (graf 1)**
- **id_seed_node_n_undirected.pickle (graf 2)**
- **id_seed_node_n_undirected_reduced.pickle (graf 3)**

indiqueu-ne les següents mesures: ordre, mida i grau mitjà dels nodes. En cas d'un graf dirigit, doneu el grau mitja d'entrada i el grau mitjà de sortida.

Graf	Ordre	Mida	Grau mitjà dels nodes	Graf dirigit	Grau mitjà d'entrada	Grau mitjà de sortida
1	40	39	1,026	Sí	1.0256410	1.0256410
2	2	1	1	No	-	-
3	1	0	0	No	-	-

Justifiqueu si el graf dirigit obtingut de l'exploració inicial del crawler pot o no pot tenir més d'una component dèbilment connexa i per què. Indiqueu quina relació té això amb la selecció d'una única llavor.

Com que el graf dirigit obtingut de l'exploració inicial del crawler és un arbre, no pot tenir més d'una component dèbilment connexa.

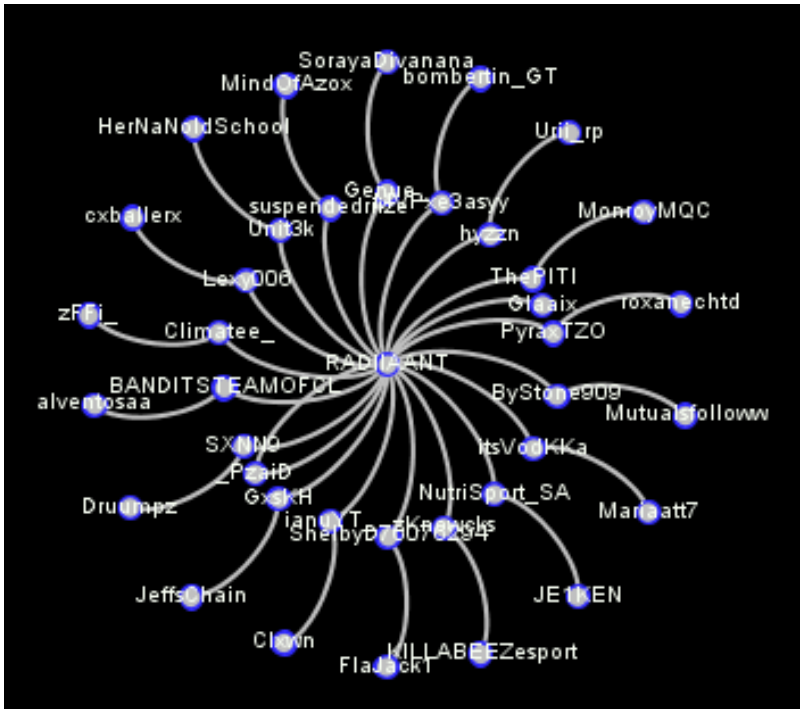
Si per altra banda seleccionéssim n nodes llavor, llavors podem arribar a tenir n components debilment connexes. També es podria donar el cas de seleccionar n nodes llavor però tenir menys de n components debilment connexes degut a que dues components inicialment dèbilment connexes s'haguessin connectat, formant-ne una de sola.

Justifiqueu també, si els grafs simètrics dels altres dos fitxers poden tenir o no més d'una component connexa.

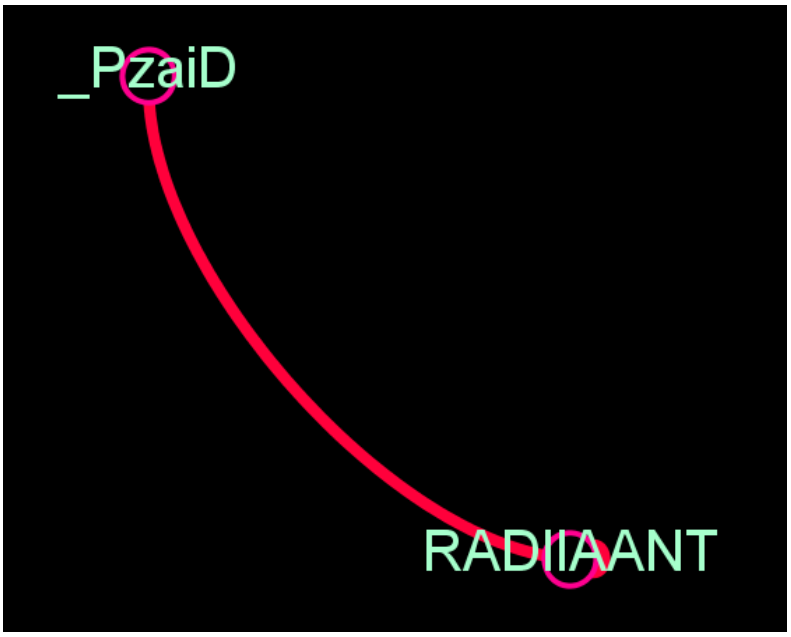
En els casos de grafs simètrics, es molt poc probable tenir components separades; es a dir que dos vèrtexs no es puguin arribar a connectar. Això com a molt es podria donar

en un cas remot en el que el crawler es quedes sense cua(per una quantitat massiva d'usuaris privats o de bots) llavors hauria de començar amb una segona arrel generant dos components connexes separats; però fora d'aquest cas inhòspit en un graf simètric com el generat per nosaltres només hi hauria d'haver una única component connexa (en el nostre cas sobre tot que es un arbre)

Utilitzant el Gephy, obriu cada un dels tres grafs que heu generat. Per fer-ho haureu d'obtenir un fitxer en format Graph Exchange XML per a cada graf, utilitzant la funció export graph to gexf(g,file name) que heu implementat en la Part 1 d'aquesta pràctica. Per a cada graf trobeu la millor visualització possible, a poder ser que mostri informació d'algunes característiques dels nodes i les arestes, i incloeu en el pdf la imatge d'aquesta visualització. Noteu que és possible que per alguns grafs la visualització sigui molt pobre. En base a aquestes visualitzacions, comenteu per què ha estat interessant obtenir els grafs id_seed_node_n_undirected i id_seed_node_n_undirected_reduced a partir del primer graf explorat. Opcionalment, podeu pensar altres sistemes de transformació del graf inicial per tal d'obtenir un altre graf més gestionable. Justifiqueu el per què d'aquestes transformacions i implementeu les funcions adients.



Graf 1



Graf 2

En aquest cas, la visualització és pobre perquè com hem comentat anteriorment, només tenim un parell de nodes que tinguin relació mútua. En el cas d'aquest algorisme, és

difícil que dos nodes tingui relació alhora, però en el cas del node llavor escollit, aconseguim una sola relació mútua.

Aquesta visualització és interessant perquè en l'algorisme permet comprovar ràpidament (pensant en si tinguéssim un graf de milers de nodes) si algun usuari del primer nivell té com a segon nivell l'usuari llavor. En aquest cas, com havíem comentat anteriorment, passa en un sol node.

Graf 3

En aquest cas, la visualització és d'un sol node perquè l'algorisme no es basa en construir més d'una aresta per cada node. Si afegim el requisit de que només volem visualitzar nodes de grau 2 mínim, ens passa que només visualitzem el node llavor.

Aquesta visualització és interessant perquè permet visualitzar els nodes que tenen grau major al grau que tu escullis. En el meu algorisme, però, aquesta representació no és gaire útil perquè sempre hi acabarà apareixen només el node llavor.

3a part: anàlisi de la informació

Determineu, proporcionant un gràfic de distribució de graus amb escala log/log, si les distribucions de graus dels tres grafs obtinguts en les dues primeres parts de la pràctica segueixen una llei de la potència, tal i com correspon a les dades d'una xarxa social.

A causa de la poca quantitat de nodes no s'apreciaria pràcticament res en representar-se però realment si que hi ha un creixement exponencial a través de les capes, de fet si a la última capa s'hagués fet els 20 més importants encara destacaria més aquesta condició. De fet si ho mires objectivament a la capa 0 el grau és 1 en la capa 1 el grau és 2 a la capa 3 el grau és 20; queda clar el creixement exponencial propi d'una xarxa social.

Suposeu que voleu llançar una campanya publicitària entre els usuaris de Twitter que teniu en el graf del fitxer id_seed_node_n_undirected_reduced.pickle i només podeu fer que 5 dels usuaris difonguin la informació. Suposant que aquest graf fos el que conté totes les connexions entre aquests usuaris, quins usuaris seleccionaríeu per tal que la informació es propagués el màxim possible? Indiqueu els nodes seleccionats i expliqueu el per què de la selecció.

Suposant que el graf del fitxer *id_seed_node_n_undirected_reduced.pickle* conté bastants usuaris i que conté totes les connexions entre aquests usuaris, escolliríem els 5 usuaris que tinguessin un grau de sortida major per a difondre informació. D'aquesta manera, aconseguiria difondre la informació al màxim nombre d'usuaris possibles i en el mínim de passos.

No s'indiquen els nodes seleccionats en el meu cas, perquè el graf del fitxer mencionat anteriorment només té 1 sol node; de la mateixa manera si agaféssim el graf no reduït s'escolliria el node llavor i els fills més importants, és a dir , que tenen més seguidors.

Prenent el graf del fitxer id_seed_node_n_undirected_reduced.pickle, trobeu els cliques de mida més gran o igual que min_size_clique. Indiqueu quin valor heu pres com a min_size_clique i el nombre total de cliques que heu trobat de cada mida.

Com ja he comentat anteriorment, el graf del fitxer *id_seed_node_n_undirected_reduced.pickle* només té 1 node, per tant, es respondrà l'exercici basant-me en el graf inicial que s'obté amb la funció *crawler*.

Un clique és un subgraf d'un graf en el que cada node està connectat a tots els altres vèrtexs del subgraf.
Com a valor de min_size_clique s'ha escollit 2, ja que amb 3 no existeix cap clique, i amb 2 s'ha tingut "la sort" d'obtenir-ne una (tot i que no es sort perquè es deu a l'elecció del node inicial).
Tenint la definició en compte, és lògic que només s'obtingui 1 clique en el nostre graf, perquè com s'ha comentat anteriorment, només un sol node del primer nivell d'exploració es relaciona amb el node llavor. Aquest node (anomenat '_PzaiD'), juntament amb el node llavor, són els dos nodes que formen l'única clique del graf.

Calculeu i indiqueu el nombre total de nodes diferents que formen part tots aquests cliques. Com creieu que hauria de ser el grau d'aquests nodes, petit o gran?
Només es té un sol clique de dos nodes. El clique està format pels nodes ['RADIIAANT', '_PzaiD']. El grau d'aquests dos nodes pot ser petit o gran, però sempre tenint en compte que com a mínim ha de ser un grau igual al nombre de components de la clique; per tant es pot dir que el grau dels nodes del subgraf vindrà donat pel tamany del clique com a mínim.

Seleccioneu un parell de nodes d'aquesta llista amb grau petit (indiqueu-ne quins), mireu la mida del clique als que pertanyen i discutiu per què tenen un grau baix en aquest graf?
Si es seleccionen els dos únics nodes que formen clique en el nostre graf no direccional, s'obté una mida 2 del clique. És trivial que tinguin un grau baix en el graf, ja que al escollir explorar només 40 nodes, no podem pretendre que si el nostre algorisme crea alguna clique (que ja és molt), els seus components tinguin un grau alt. Si suposem que el graf amb el que treballem es dirigit de per si; podem obtenir molts cliques de grau 1, que serien les de grau més petit i seria així degut a que serien les fulles de l'arbre obtingut i com a tals només estarien connectats als seus pares. Com a exemple podríem agafar els subgrafs generats per les arestes que connecten la primera capa amb la segona.

Trieu un dels cliques de mida màxima i analitzeu els comptes de Twitter que en formen part. Intenteu trobar algun tret que defineixi aquesta comunitat i expliqueu-lo.

Si s'hagués explorat més de 40 nodes i al complet, s'haurien trobat més cliques i de més components. En aquest cas, cadascuna de les comunitats podria ser un equip de competitiu (amb els seus jugadors, els reserves i el mànager) , podem trobar com a comunitat un conjunt de jugadors professionals sobre el mateix joc; també sobre el tipus de joc(guerra, conducció…) o directament unes comunitats de gent que els hi agrada els videojocs. Tot depèn de la quantitat de nodes llavor i de la quantitat de nodes a explorar.

Trobeu el màxim k-core del graf del fitxer_id_seed_node_n_undirected_reduced.pickle, és a dir el valor k màxim per al qual el k-core no és un graf buit. Indiqueu el valor de k i l'ordre i la mida del k-core en qüestió.

Es defineix k-core com el subgraf maximal inclòs en un altre graf en el que tots els nodes d'aquest subgraf tenen com a mínim grau k.

Tal i com ha passat en un exercici anterior, el graf del nostre fitxer *id_seed_node_n_undirected_reduced.pickle* només té 1 node, per tant, es respondrà també a l'exercici basant-nos en el graf inicial que s'obté amb la funció *crawler*.

id_seed_node_n_undirected_reduced.pickle

Valor de k	Ordre k-core	Mida k-core
0	**1**	**0**

id_seed_node_n

Valor de k	Ordre k-core	Mida k-core
1	**39**	**40**

En l'obtenció de graf que heu realitzat a la primera part de la pràctica, només heu guardat les arestes que us han anat apareixent al crawling però no s'indicava de guardar cap atribut del node. En particular, amb el graf obtingut no teniu indicat si un node ha estat explorat o descobert i tampoc teniu identificat el node llavor del crawler. Indiqueu quines mètriques podríeu utilitzar per intentar classificar els nodes entre explorats i descoberts. Justifiqueu el perquè de la vostra resposta en funció del sistema de crawling que hagueu implementat.

Per classificar els nodes entre explorats i descoberts es podria haver utilitzat dos arrays dinàmics, implementats a python a través de les llistes, un de nodes explorats i l'altre de nodes descoberts. D'aquesta manera, la llista de nodes descoberts anirà augmentant cada vegada que es descobrís un nou node. Aquests nodes, per ordre, s'aniran recorrent i afegint a la llista de nodes explorats un a un, cada vegada que s'exploressin. Un cop finalitzat l'algorisme, tindrem les dues llistes: en la de nodes explorats n'hi hauria tants com el màxim nombre de nodes que volíem explorar passat com a paràmetre de la funció crawler (en aquest cas 40). En la llista de nodes descoberts, tindríem una quantitat bastant elevada de nodes (entre ells, els nodes explorats, evidentment).

En el nostre algorisme en concret, es poden tenir aquestes dues llistes (nodes explorats i nodes descoberts) per a cada nivell d'exploració per a facilitar i diversificar la feina. Quan un nivell d'exploració es donen per acabat, les llistes passaran a ser dues llistes totalment noves, gravant-se la informació a disc i descarregant força de la memòria RAM.

De la mateixa manera que podem guardar la informació en dues llistes, també és es podria fer en hash maps, diccionaris en python, per tal de guardar més informació que les connexions en si; podem saber si estan connectats o no a certes pàgines per fer una classificació manual prèvia, podríem també afegir la raó entre seguidors i amics per considerar la seva "popularitat" o fins i tot , i parlant de paraules majors, podem entrenar una intel·ligència artificial per ser capaç de fer certes classificacions identificades amb un codi i anar afegint aquests codis al diccionari amb la clau del node per poder-los anar classificant durant el crawling . Això podria ser de molta utilitat per detectar subcomunitats en una comunitat fàcilment diferenciable

EXTRA

Veient la alta potencia que té gephi hem intentat pensar com explotar al màxim tant les seves capacitats de visualització com la facilitat amb la que ens generar gràfiques i càlculs; per tant s'ha pensat que es podria intentar fer un script que permet l'edició i càlculs sobre gephi en massa.

S'ha fet servir la llibreria pyautogui de python per tal de poder accedir a l'automatització del teclat i el mouse sobre el computador; actualment el script està dissenyat per funcionar en un dels dos ordinadors del grup degut a la complexitat de generalitzar a diferents sistemes operatius. Aquest script seria funcional sobre qualsevol sistema Linux semblant a Ubuntu; només s'haurien de modificar certs paths.

Així doncs aquest script tindria com a algoritme rebre nom d'usuari i paths fins a gephi i els diferents grafs com a input i començar a simular la interacció d'un usuari amb l'ordinador. Principalment el que fa es obrir terminal, moure's fins a l'executable del gephi, obrir el gephi, fer tots els càlculs que gephi permet i descarregar el csv amb tota la informació per si després es vol automatitzar un procés sobre la informació en un format més tangible.

S'adjunta l'script (no genèric, de mostra per un graf en un ordinador) i un vídeo que mostra el seu funcionament(el vídeo s'ha hagut de gravar des d'un dispositiu mòbil per la dificultat de trobar un screen-recorder gratuït que no pixeli l imatge).

Fins i tot per a fer més pràctica el que seria la funcionalitat del projecte se'ns va acudir fer una aplicació de java que rebés com a inputs els mateixos que el crawler de la part 1 del projecte i escriure els càlculs que es vol que posteriorment es facin amb gephi. Desgraciadament per manca de temps ens hem quedat només en el disseny gràfic de l'aplicació, més que res per el problema que ha donat java per interactuar amb python a través de la terminal.