2077: La privacitat de les connexions dels usuaris d'una xarxa social

Memòria del projecte de final de carrera corresponent als estudis d'Enginyeria Superior en Informàtica presentat per Cristina Pérez Solà i dirigit per Jordi Herrera Joancomartí.

Bellaterra, juny de 2010

El firmant, Jordi Herrera Joancomartí, professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Cristina Pérez Solà

Bellaterra, juny de 2010

Firmat: Jordi Herrera Joancomartí

Als meus pares, per donar-me l'oportunitat d'arribar fins aquí.

Agraïments

Especialment, al Jordi Herrera, per les seves innumerables correccions, el seu suport constant i per donar-me l'oportunitat d'entrar a formar part del departament, al qual m'agradaria estendre també el meu agraïment.

Al César Córcoles, per renunciar a una mica de la seva privacitat i permetre'ns fer-lo servir com a usuari inicial de les exploracions.

Als meus amics, per suportar-me dia rere dia.

Índex

Índex de figures

CAPÍTOL 1

Introducció

1.1 Motivacions

En els últims anys hi ha hagut un gran augment de l'ús de les anomenades xarxes socials *online*. Aquests serveis, que construeixen comunitats *online* centrades en les relacions entre usuaris, suposen nous riscos per a la privacitat de qui els fa servir, alhora que ofereixen l'oportunitat d'estudiar les relacions socials de manera molt acurada.

L'anàlisi de les xarxes socials ha estat des dels seus inicis pluridisciplinari. Sociòlegs, epidemiòlegs i economistes, entre d'altres, n'han estudiat el seu comportament des de fa aproximadament un segle. Aquest estudi però, es trobava força limitat per la dificultat de recol·lectar dades amb les quals treballar. Un dels experiments més famosos sobre xarxes socials va ser el conegut com a experiment dels sis graus de separació [47] (tot i que Milgram, el seu creador, mai es va referir a ell amb aquest nom). L'experiment pretenia mesurar la llargada mitjana del camí que separa dos habitants dels Estats Units. Per fer-ho, Milgram seleccionava alguns individus de la població dels estats del centre i els enviava una carta detallant l'objectiu de l'experiment: aconseguir fer arribar la mateixa carta a un destinatari especificat. Si la persona que rebia la carta coneixia personalment al destinatari que s'esmentava a la carta, aleshores podia lliurar-li directament. En cas contrari, calia donar la carta a un conegut que creguessin que es trobava més proper al destinatari, havent signat la carta prèviament. D'aquesta manera, quan la carta arribava al destinatari final, aquesta contenia una llista de les persones que l'havien transmès. La única font de dades de l'experiment eren aquestes cartes que, una vegada arribessin al seu objectiu, havien de ser enviades als investigadors per al seu anàlisi.

Amb l'aparició de les xarxes socials *online*, tot això canvia radicalment. Per primera vegada, es disposen de grans quantitats de dades que expressen relacions entre individus. A més, aquestes dades es poden obtenir sovint de fonts públiques, el que les fa accessibles per al seu estudi. És en aquest context quan les ciències de la computació i la informació se sumen a la llista de disciplines que estudien les xarxes socials. Tal com veurem més endavant, per estudiar les xarxes socials aquestes s'acostumen a representar en forma de graf. Els grafs que representen una xarxa social són anomenats grafs socials.

Actualment, les xarxes socials *online* són un fenomen molt estès. Hi ha molts llocs web que ofereixen aquests serveis i milions d'usuaris que en fan ús. Ens trobem doncs, amb un gran nombre de xarxes socials *online*, cadascuna de les quals posa a disposició pública informació sobre els usuaris que la formen i les seves relacions. La informació que podem extreure d'una xarxa social *online* pot no ser suficient per conèixer totes les relacions entre els individus d'una població concreta: ens podem trobar amb individus que no formen part d'aquesta xarxa social, o bé, que en formen part però que no tenen totes les relacions explicitades. A més a més, hi haurà informació que existeix a la xarxa social *online* però que no som capaços d'extreure. Per aquest motiu, el coneixement que tindrem sobre les relacions entre els individus serà parcial.

Aquest projecte se centrarà en l'anàlisi de les dades que ofereixen diferents xarxes socials *online* amb l'objectiu principal d'agregar la informació obtinguda de diferents fonts per tal d'intentar millorar el coneixement global que es té sobre les relacions entre individus.

1.2 Objectius

L'objectiu principal d'aquest projecte és l'obtenció de les relacions socials d'una persona o comunitat utilitzant agregació d'informació pública de diferents xarxes socials *online*. Per tal d'assolir aquest objectiu, caldrà dividir-lo en una sèrie de subobjectius, que alhora serviran per definir les diferents tasques que conformaran el projecte.

En primer lloc, caldrà analitzar les propietats dels grafs que representen xarxes socials. Conèixer amb quin tipus de graf s'està tractant permetrà optimitzar la recollida de dades així com el seu posterior anàlisi. De la mateixa manera, també caldrà analitzar les tècniques de *graph-mining* [14] que s'utilitzen per estudiar les xarxes socials.

Per tal de poder realitzar l'agregació d'informació de diferents fonts serà necessari haver obtingut anteriorment la informació de cadascuna de les fonts. Per aquest motiu, un dels subobjectius més crític serà la recol·lecció d'informació de diferents fonts i la creació dels grafs socials que se'n derivin. Aquest subobjectiu es pot alhora dividir en dos: l'estudi dels mètodes de *web-crawling* existents i l'elaboració del programari que realitzi l'extracció de dades.

Una vegada obtinguda la informació de diferents fonts, caldrà buscar correspondències en

cada una de les xarxes analitzades. Per fer-ho, serà necessari estudiar la relació entre els grafs socials obtinguts de les diverses fonts i els algorismes que permeten establir coincidències entre els diferents grafs. S'implementaran alguns d'aquests algorismes per tal d'aconseguir l'objectiu principal.

Una vegada assolit l'objectiu principal, serà interessant tenir alguna manera de quantificar la millora global de la informació obtinguda respecte a cada una de les fonts utilitzades. Per tant, caldrà definir alguna mesura que ens permeti aquesta quantificació.

1.3 Anàlisi de viabilitat tècnica

Com s'ha vist anteriorment, el projecte consta de dues parts diferenciades: la recol·lecció de dades i l'agregació de les mateixes.

Per tal de construir un graf social caldrà recol·lectar dades sobre els usuaris i les seves connexions de la xarxa social analitzada. La informació que es vol recol·lectar és informació pública a la majoria de xarxes socials *online*. Per tal de poder realitzar aquesta recol·lecció, serà necessari disposar d'accés a la xarxa social *online* a analitzar (accés a Internet). Atès que les quantitats de dades amb les que es treballaran són elevades, caldrà també disposar d'algun mecanisme de recol·lecció automàtica de dades. Diversos llenguatges de programació ofereixen classes que permeten programar la recol·lecció de dades de manera senzilla.

La segona part del projecte presenta una major complexitat tècnica. Com es veurà posteriorment, els algorismes de correspondència inexacta de grafs tenen complexitats elevades. Per tal de dur a terme aquesta part del projecte, s'implementarà un algorisme similar al proposat a [35]. Els resultats mostrats a l'article demostren que tant la seva implementació com la seva utilització per al problema presentat són viables a nivell tècnic.

1.4 Planificació inicial

Les diferents tasques que formen aquest projecte es troben estretament relacionades amb cada un dels objectius. Així, tindrem sis tasques diferenciades, algunes de les quals es trobaran dividides en diferents subtasques:

1. Estudi de les propietats dels grafs socials i de les tècniques d'anàlisi dels mateixos
2. Recol·lecció de dades
   (a) Anàlisi de les tècniques de *web-crawling* existents
   (b) Elaboració d'un programari modular i parametritzable per a la recol·lecció de dades

i. Anàlisi, especificació i disseny
ii. Desenvolupament
iii. Test

3. Agregació d'informació
   (a) Representació de grafs i implementació d'algorismes bàsics
   (b) Implementació d'algorismes que realitzin *graph-matching* inexacte entre diferents grafs
4. Experimentació amb les dades recollides i els algorismes implementats
5. Quantificació de la millora d'informació respecte cada una de les fonts
6. Elaboració de la documentació (memòria i presentació)

El diagrama de Gannt de la planificació inicial es troba a la Figura 1.1.

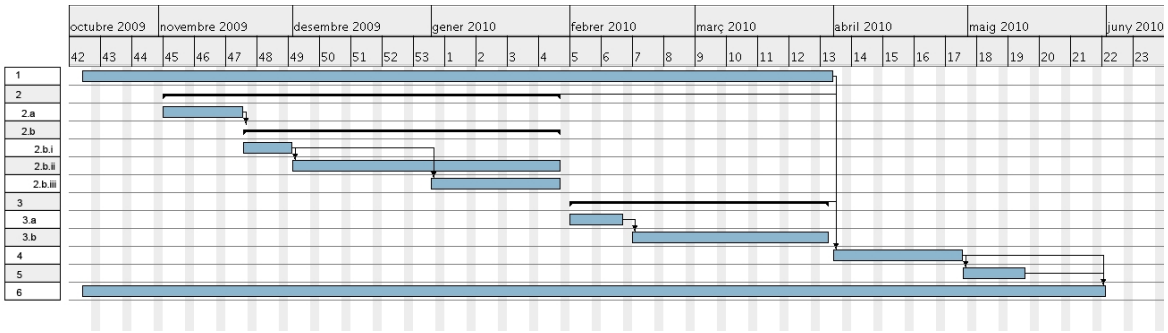

Figura 1.1: Diagrama de Gannt.

1.5 Estructura de la memòria

Aquesta memòria es troba estructurada en 7 capítols: introducció, conceptes bàsics, anàlisi de la informació, aplicacions, anàlisi de les dades obtingudes i conclusió.

1. El capítol que esteu llegint és la introducció, on s'ha explicat quins són els objectius d'aquest projecte, les motivacions per dur-lo a terme, un petit estudi de viabilitat, la planificació inicial del projecte així com l'estructura d'aquesta memòria.
2. El segon capítol és un recull dels conceptes bàsics necessaris per entendre la resta de la memòria. El capítol comença amb algunes definicions bàsiques sobre grafs que es faran servir a la resta de capítols. Seguidament, trobem una introducció a les xarxes socials on es descriu què són els grafs socials, quina estructura tenen, com obtenir aquests grafs i quins riscos suposen per a la privacitat.

3. El tercer capítol se centra en el procés de recollida d'informació, explicant quines fonts d'informació es faran servir, com es procedirà a obtenir la informació de les fonts, com s'emmagatzemarà aquesta informació i com s'exportarà posteriorment per tal de poder-la visualitzar en programes específics d'anàlisi i visualització de grafs.

4. Una vegada obtinguda la informació, es procedirà a analitzar-la. El quart capítol se centra en aquest anàlisi de la informació, que constarà, principalment, de dues fases. En primer lloc, es presenten algunes de les mètriques que es fan servir per analitzar grafs. En segon lloc, s'explica la tècnica d'agregació d'informació que es farà servir per tal d'agregar grafs.

5. El cinquè capítol presenta les aplicacions desenvolupades i utilitzades en les diferents fases d'aquest projecte. El capítol està dividit en tres seccions, corresponents a cada una de les fases de tractament d'informació: recollida, representació i anàlisi.

6. El sisè capítol presenta les dades recollides seguint els procediments explicats al Capítol 3 i les analitza aplicant les mesures explicades al Capítol 4, fent servir les aplicacions presentades al Capítol 5. El capítol s'estructura en tres seccions: un resum de les dades recollides, l'anàlisi de les dades i l'aplicació del procés d'agregació.

7. L'últim capítol està reservat a les conclusions, on veurem un resum del que ha estat el desenvolupament del projecte, possibles millores i línies d'investigació obertes, a més de les conclusions pròpiament dites.

CAPÍTOL 2

Conceptes bàsics

Aquest capítol és un recull dels conceptes bàsics que es desenvoluparan a la resta de capítols. El capítol comença amb un repàs de terminologia sobre teoria de grafs. Tot seguit, s'exposa què són les xarxes socials, quines propietats tenen els grafs socials i com podem obtenir els grafs socials. Per últim, es comenten els riscos que poden suposar les xarxes socials per a la privacitat dels seus usuaris.

2.1 Grafs

Un **graf** $G = (V, E)$ està format per un conjunt no buit d'elements V i per un conjunt E de parells no ordenats d'elements diferents de V.

Els elements de V els anomenarem **vèrtexs** o **nodes** i els elements d'E **arestes**. Si $e = (u, v)$ és una aresta, aleshores direm que u i v són vèrtexs **adjacents** i que l'aresta e és **incident** als nodes u i v.

Donat un graf $G = (V, E)$, si existeixen dues o més arestes $a, b \in E$ tals que $a = (u, v)$ i $b = (u, v)$ aleshores direm que G és un **multigraf**.

El concepte de **digraf** o **graf dirigit** deriva directament del concepte de graf exigint que les arestes, que ara en direm **arcs**, siguin parells ordenats de vèrtexs diferents. Un digraf o graf dirigit $G = (V, E)$ està format per un conjunt no buit V i per un conjunt E de parells ordenats d'elements diferents de V.

En un graf dirigit distingirem entre **successors** Γ i **antecessors** Γ^{-1} d'un vèrtex. Definim el conjunt de successors de v com a $\Gamma_v = \{u \in V \text{ t.q. } \exists\ e \in E \text{ amb } e = (v, u)\}$. De la mateixa manera, definim el conjunt d'antecessors de v com a $\Gamma_v^{-1} = \{u \in V \text{ t.q. } \exists\ e \in E \text{ amb } e = (u, v)\}$.

El graf **subjacent** d'un graf dirigit és el graf que s'obté quan prescindim de l'orientació de les arestes. Podem veure un exemple d'un graf dirigit i el seu corresponent graf subjacent a la Figura 2.1.

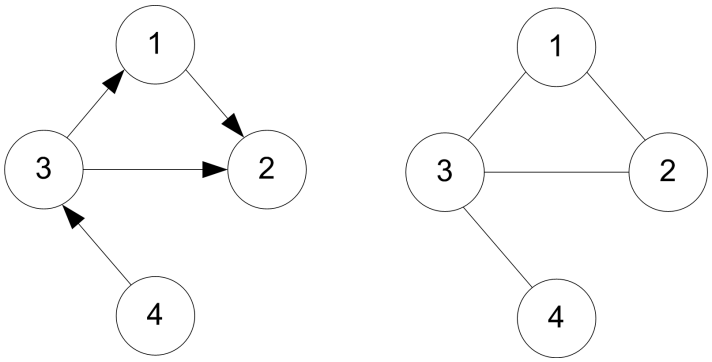

Figura 2.1: Un graf dirigit (esquerra) i el seu corresponent graf subjacent (dreta).

L'**ordre** d'un graf $G = (V, E)$ és el nombre de vèrtexs de G, és a dir, el cardinal de V que denotarem per $|V|$. La **mida** de G és el nombre d'arestes de G, és a dir, $|E|$. Aquests conceptes són anàlegs per als digrafs.

El **grau** d'un vèrtex v ($grau(v)$) és el número d'arestes incidents a aquell vèrtex. En els digrafs, distingirem entre el **grau interior** i el **grau exterior** d'un vèrtex. El grau interior d'un vèrtex és el nombre de antecessors mentre que el grau exterior d'un vèrtex és el nombre de successors.

El **grau mitjà** d'un graf és el resultat d'aplicar la mitjana aritmètica dels graus de tots els seus nodes (o bé $\frac{2|E|}{|V|}$).

Un graf on tots els nodes tenen el mateix grau k s'anomena graf **k-regular**. Un graf k-regular amb $k = |V| - 1$ s'anomena graf **complet**.

Un **camí** és una seqüència de vèrtexs on cada parell de vèrtexs consecutius són adjacents. Els camins poden ser infinits. Si un camí és finit, aleshores té un vèrtex d'origen i un vèrtex final. Els vèrtexs d'origen i final s'anomenen **vèrtexs terminals**. La resta de vèrtexs d'un camí són **vèrtexs interns**. El camí que compleix que el seu vèrtex d'origen i de final és el mateix és anomenat **cicle**.

Un camí que no conté vèrtexs repetits és un **camí simple**. Anàlogament, un cicle que no conté vèrtexs repetits (excepte la necessària repetició de l'origen i el final) és un cicle simple. Un cicle simple que conté tots els vèrtexs d'un graf és un **cicle Hamiltonià**.

La **longitud** d'un camí és el nombre d'arestes que conté, mentre que la **distància** entre dos vèrtexs d'un graf és la longitud mínima dels camins que els connecten. Un camí geodèsic (o una **geodèsica**) és el camí més curt (de distància mínima) entre dos nodes. Hi pot haver més d'una geodèsica per un mateix parell de nodes.

Anomenem **component connex** d'un graf a un conjunt maximal de nodes tal que existeix almenys un camí entre cada parell de nodes. Un graf amb un sol component connex s'anomena graf connex. En el cas dels digrafs, si s'exigeix la connexió dels parells de nodes en els dos sentits, s'anomena **component fortament connex**.

El **conjunt dominant** d'un graf $G = (V, E)$ és un subconjunt $V' \subseteq V$ tal que cada vèrtex $v \in V$ que no pertanyi a V' és adjacent a almenys un vèrtex de V'. El **número dominant** $\gamma(G)$ és el cardinal del menor conjunt dominant de G. Podem veure dos exemples de diferents conjunts dominants d'un mateix graf a la Figura 2.2.

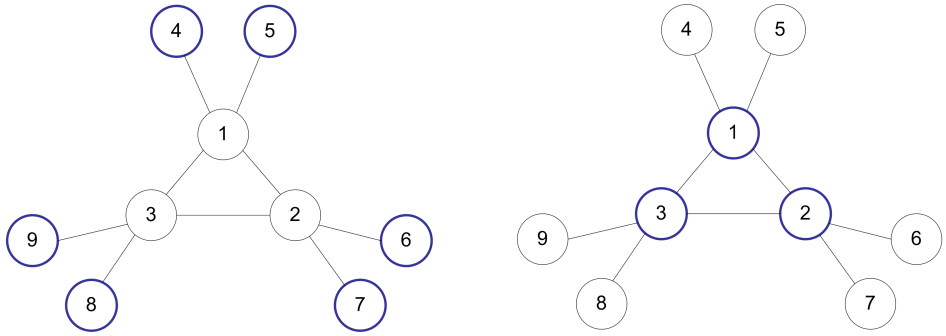

Figura 2.2: Dos exemples de conjunts dominants. El conjunt dominant marcat al graf de la dreta és el menor conjunt dominant del graf ($\gamma(G) = 3$).

El subgraf **induït** per un conjunt de vèrtexs $V' \subseteq V$ és el subgraf $G' = (V', E')$ tal que $E' = \{(u, v) \in E \mid u, v \in V'\}$.

2.1.1 Notació

Per tal de no repetir declaracions al llarg d'aquesta memòria, s'usaran les següents convencions a partir d'aquest punt.

Donat un graf G, considerarem que $G = (V, E)$ amb V el conjunt de vèrtexs de G i E el conjunt d'arestes.

Anomenarem n al nombre de vèrtexs del graf ($n = |V|$).

Donat un vèrtex $v \in G$, anomenarem $grau(v)$ o $dg(v)$ al seu grau, és a dir $grau(v) = |N(v)|$ amb $N(v) = \{u \in V | (v, u) \in E\}$.

2.2 Xarxes socials

Una xarxa social és una estructura social constituïda per individus (o entitats) que es troben interconnectats a través de diferents tipus de relacions.

Les xarxes socials *online* (*Online Social Networks* o OSN) són serveis web que permeten que els usuaris

1. elaborin un perfil públic, o parcialment públic, que els descriu
2. explicitin les seves relacions amb altres usuaris
3. comparteixin la informació sobre les seves relacions amb altres usuaris de la xarxa

A nivell general, el funcionament d'una OSN és el següent. En primer lloc, els usuaris es donen d'alta tot omplint un formulari amb informació personal que pot incloure des del nom, la població de residència, l'edat o les aficions fins a fotografies, vídeos o altres tipus de contingut multimèdia. Aquesta informació formarà part del perfil de l'usuari. La visibilitat d'aquest perfil dependrà tant de la xarxa com de les preferències de l'usuari. Algunes OSN com ara *Friendster* o *Tribe.net* defineixen els perfils dels usuaris com a públics per defecte i permeten que siguin indexats pels motors de cerca, fent-los accessibles per qualsevol persona, independentment que aquesta disposi d'un compte a l'OSN en qüestió. En canvi, *LinkedIn* defineix la informació que pot veure cada un dels seus usuaris segons si disposen o no de comptes de pagament. Altres OSN defineixen diversos grups amb permisos diferents sobre els perfils. És el cas de *Facebook*, que permet configurar la visibilitat del perfil depenent de la posició de l'usuari a la xarxa. Així, un perfil pot ser públic (visible per a tothom), visible només per als membres del mateix grup o visible només per als amics. Altres esquemes híbrids o alternatius són utilitzats per diverses OSN. La manera com es gestiona la visibilitat de la informació dels perfils dels usuaris és una característica diferenciadora entre les OSN.

En segon lloc, els usuaris identifiquen a altres usuaris amb els quals tenen algun tipus de relació. Les etiquetes que s'assignen a aquestes relacions i que permeten definir-les varien d'una OSN a una altra. "Amic", "contacte", "fan" o "seguidor" són algunes de les etiquetes més utilitzades. Algunes OSN exigeixen que les relacions especificades siguin bidireccionals, necessitant la confirmació dels dos usuaris involucrats per ser creades. És el cas, per exemple, de *Facebook*, que requereix que les relacions d'amistat siguin confirmades per les dues parts abans de ser creades. En canvi, altres OSN estableixen relacions unidireccionals. És el cas, per exemple, de *Twitter*, que permet establir relacions de seguiment no recíproques.

A part de crear perfils i relacions i compartir aquesta informació, les OSN acostumen a incloure altres serveis com ara missatgeria instantània, serveis de *microblogging*, compartició d'arxius multimèdia o jocs *online*.

La característica principal que fa de les OSN una font d'interès per a investigadors de diverses àrees és el fet de permetre articular i fer visibles les xarxes socials. Mentre que les relacions

en una xarxa social són difícils de veure i recol·lectar, en una OSN les relacions queden totalment definides pels seus usuaris i, encara més, sovint són fetes públiques. D'aquesta manera, l'anàlisi de les OSN permet disposar d'un conjunt de dades molt gran i complet sobre el qual treballar.

2.2.1 Grafs socials

Els grafs socials són grafs que expliciten una xarxa social, de manera que els nodes representen individus i les arestes les seves relacions. Les relacions representades en un graf social poden ser molt diverses: econòmiques, sentimentals, d'amistat, de col·laboració acadèmica... El tipus de relació que representin les arestes determinarà en gran mesura si el graf resultant és, o no, un graf dirigit. Per exemple, per a una relació sentimental, el graf resultant no serà dirigit: si l'Alice està mantenint una relació sentimental amb en Bob, en Bob també es trobarà en una relació amb l'Alice, pel que no serà necessari que el graf sigui dirigit. En canvi, si es representa una relació econòmica, per exemple, els diners que es deuen diferents individus, aleshores serà necessari que el graf sigui dirigit: caldrà distingir entre que l'Alice degui 100 euros a en Bob i que en Bob degui 100 euros a l'Alice. Hi haurà cert tipus de relacions, com ara l'amistat (en el sentit en el que es fa servir en les xarxes socials *online*), que permetran els dos tipus de representació. Representar l'amistat com un graf no dirigit suposarà haver d'emmagatzemar menys dades i facilitarà els càlculs sobre el graf. En canvi, representar-la amb un graf dirigit, ens permetrà obtenir mesures més precises. Per exemple, podrem distingir entre la popularitat d'una persona (el nombre d'arestes incidents al node) i la seva sociabilitat (nombre d'arestes que surten del node).

La representació de les xarxes socials a través de grafs permet analitzar-les de manera sistemàtica amb tot el conjunt d'eines matemàtiques d'anàlisi de grafs. El conjunt de tècniques que es fan servir per analitzar els grafs que representen les xarxes socials és conegut com a *Social Network Analysis* (SNA).

2.2.2 Estructura dels grafs socials

Els grafs socials reuneixen un conjunt de característiques que els distingeixen d'altres grafs. Una de les característiques més destacades és la distribució dels graus dels nodes en una llei de la potència (*power law*).

Diem que dues variables x i y es troben relacionades per una llei de la potència quan

$$y = f(x) = Ax^{-\lambda} \tag{2.1}$$

amb A i λ constants positives. λ es coneix com a exponent de la llei de la potència.

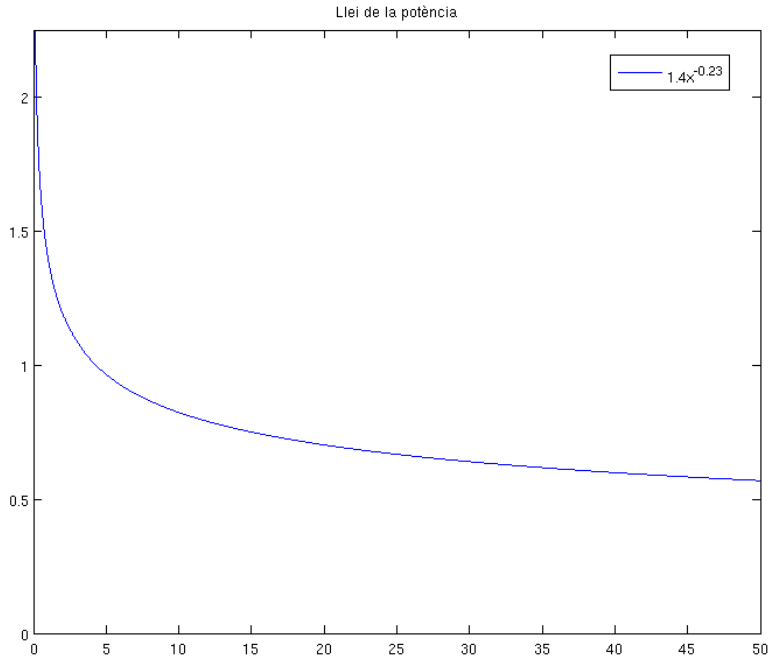

Figura 2.3: Exemple de llei de la potència amb $A = 1.4$ i $\lambda = 0.23$.

Per tant, els grafs socials es caracteritzen per tenir pocs nodes de grau molt elevat i una gran quantitat de nodes amb un grau petit.

Si la distribució dels graus dels nodes d'un graf segueix una llei de la potència, es diu que el graf és invariant respecte l'escala ja que, donada una relació $f(x) = Ax^{\lambda}$, multiplicar l'argument x per una constant només produeix un canvi proporcional al resultat de la funció $(f(Kx) = A(Kx)^{\lambda} = bf(x))$.

A més de la distribució dels graus dels nodes d'un graf social, les *power laws* governen una gran varietat de fenòmens: la topologia d'Internet [21], les citacions en literatura científica [38], les freqüències d'ús de les paraules en els idiomes [52], la mida dels terratrèmols [23], així com la mida dels cràters de la lluna [4], en són alguns exemples.

En els grafs socials les *power laws* no són normalment visibles en conjunts de dades petits [14]. De fet, un subgraf d'un graf invariant d'escala creat a partir d'una selecció aleatòria de nodes (on cada node és inclòs en el subgraf amb una probabilitat p i deixat fora amb una probabilitat $(1-p)$) no manté la distribució de graus dels nodes (no és invariant d'escala)[45]. A més, les distribucions dels graus dels nodes poden presentar desviacions respecte a *power laws* pures, el que en complica el seu ajust. Al Capítol 6 s'analitzaran les dades recollides en aquest treball per tal de comprovar si són suficients per poder-hi ajustar una *power law*.

En el Capítol 4 analitzarem altres característiques estructurals dels grafs socials.

2.2.3 Obtenció de grafs socials

La informació que emmagatzema una OSN pot ser obtinguda de diverses maneres. La més immediata és la recol·lecció de la informació pública amb l'ús d'eines automatitzades com ara *crawlers* (aquesta serà la font d'informació per aquest projecte). Tot i així, aquest no és l'únic mètode. Les pròpies OSN comparteixen sovint aquesta informació amb terceres parts per a la realització d'estudis o bé amb finalitats comercials. La informació cedida pot comprendre tant la informació pública dels usuaris com informació amb restriccions de visibilitat o informació privada. Per aquest motiu, abans de lliurar el graf social complet, aquest passa per un procés d'anonimització amb l'objectiu d'evitar comprometre la privacitat dels usuaris.

Quan s'anonimitzen conjunts de dades relacionals s'acostuma a eliminar o xifrar els identificadors dels individus com ara noms, adreces o telèfons. Aquest procés de desidentificació, però, no garanteix que les dades hagin estat anonimitzades. Sovint aquests conjunts de dades inclouen informació com ara la data de naixement, la població o el sexe que permeten reidentificar els subjectes. Per aquest motiu, els processos de desanonimització inclouen també altres mesures com ara la introducció de soroll. Aquesta mesura consisteix en modificar alguns dels atributs publicats per tal d'evitar que les tuples siguin recognoscibles, tot mantenint les propietats estadístiques de les dades. Aquest enfoc ha estat usat àmpliament tot i que suposa un problema a l'hora d'utilitzar les dades per a estudis on no només les propietats del conjunt de dades són importants sinó també les de cada una de les tuples. En aquests casos, s'apliquen tècniques basades en el k-anonimat, consistents en realitzar transformacions a les dades per tal que cada combinació d'alguns camps concrets (que són anomenats, quasi-identificadors) coincideixi, com a mínim, en k individus diferents. Les transformacions realitzades en aquesta tècnica han de mantenir la veracitat de les dades, motiu pel qual s'acostuma a utilitzar la generalització.

Quan s'anonimitzen grafs, s'apliquen variants d'aquestes mateixes tècniques que tenen en compte no només els atributs dels nodes sinó també les estructures dels grafs en les que es troben. Aquestes tècniques intenten evitar que els nodes puguin ser reidentificats tant a partir dels seus atributs com de l'estructura del graf. El principal problema que presenta l'anonimització de grafs és que la satisfacció de les condicions necessàries per poder anonimitzar l'estructura del graf és complexa i, sovint, no és viable quan els grafs són grans o presenten un grau mitjà molt elevat. L'elaboració de tècniques que permetin aquesta anonimització és encara un problema obert sobre el qual s'està treballant actualment.

Donada l'escassetat de tècniques viables per realitzar bones anonimitzacions de grafs (potser unida a la poca preocupació per la protecció de la privacitat dels usuaris), se segueixen publicant grafs anonimitzats amb el tradicional mètode de xifrar o eliminar els identificadors. Els grafs anonimitzats seguint aquest mètode, són susceptibles a alguns atacs de desanonimització. Backstrom *et al* presenten alguns d'aquests atacs a [3]. En concret, presenten dos atacs actius (que impliquen que l'atacant modifiqui la xarxa abans que sigui desanonimitzada) i un de passiu. Els atacs actius es basen en afegir els nodes i les connexions necessàries per

tal de crear un subgraf que sigui fàcilment identificable (sigui computacionalment eficient de trobar en el graf general) i que permeti identificar els individus que es volen desanonimitzar. Una vegada s'obté el graf anonimitzat, es busca aquest subgraf que s'ha creat prèviament. D'aquesta manera s'aconsegueix identificar els individus que es volien atacar dins del graf general. Els atacs passius intenten arribar als mateixos resultats sense modificar prèviament la xarxa. En l'atac passiu descrit a [3], un conjunt d'usuaris de la xarxa social són capaços de localitzar-se a sí mateixos en el graf anonimitzat, fent servir la informació de l'estructura de la xarxa que els envolta. Tots els atacs descrits a [3] són viables, però només serveixen per atacar petits conjunts d'individus.

Narayanan i Shmatikov a [35] presenten un atac passiu aplicable a més gran escala. Aquest atac consta de dues fases: una fase d'inicialització i una de propagació. La fase d'inicialització consisteix en la identificació d'alguns nodes presents en els dos grafs que serviran de llavors per a la següent fase. Els autors proposen diverses alternatives per localitzar aquestes llavors, algunes de les quals són les comentades a [3]. Una vegada realitzada aquesta primera fase, s'obté un conjunt reduït de nodes llavor. La fase de propagació consisteix en trobar, de manera iterativa, noves correspondències a partir de les correspondències trobades a la fase anterior (a la primera iteració, a partir de les llavors de la fase inicial). Les correspondències es troben aplicant una funció heurística que ens permet valorar si dos nodes pertanyen al mateix individu. Una descripció més acurada d'aquesta tècnica es presenta a la Secció 4.2.

2.2.4 Riscos per a la privacitat

Les xarxes socials *online* posen a disposició pública una gran quantitat de dades personals dels usuaris que les utilitzen. Tot i que aquestes dades són normalment compartides pel propi usuari, sovint aquest no és conscient de fins a quin punt s'està comprometent la seva privacitat.

A més de la compartició de les dades personals dels usuaris (atributs dels nodes dels grafs socials), les OSN introdueixen una nova font de dades susceptible: les dades de les relacions (atributs de les arestes o les arestes en sí mateixes). La privacitat d'aquestes relacions és més difícil de garantir ja que surt del control del propi usuari. A més, aquestes relacions ofereixen informació molt valuosa que permet realitzar atacs de desanonimització impensables per a dades relacionals.

Les intencions per les que es pot voler recol·lectar les dades d'una OSN són moltes i molt diverses.

- *Publicitat:* Les empreses poden aprofitar la informació dels perfils d'usuari per realitzar publicitat dirigida, anant un pas més enllà de la publicitat contextual/de continguts amb la introducció d'anuncis personalitzats. La publicitat dirigida permet oferir a l'usuari anuncis adequats a la seva edat, localització geogràfica, sexe o professió. Amb

la informació inclosa en perfils d'OSN com *Lastfm* o *Flickr*, fins i tot és possible adaptar els continguts publicitaris a les preferències musicals o fotogràfiques dels usuaris.

Un altre dels usos de la informació de les OSN en publicitat és la selecció d'usuaris que es troben en punts estratègics per tal de difondre publicitat a través d'ells. Nodes que es troben molt ben comunicats[1] poden rebre publicitat amb la idea que la campanya arribarà a la resta de nodes a través d'ells. Així, per exemple, focalitzant una campanya publicitària a un conjunt de nodes dominants d'una xarxa social, es pot assegurar que tots els usuaris de la xarxa tenen almenys un amic que ha rebut la publicitat. D'aquesta manera, s'aconsegueix arribar a un major nombre d'usuaris amb un menor cost econòmic.

- *Cossos policials:* Atès que els grafs socials permeten analitzar el flux d'informació entre un conjunt d'individus, aquests són una font molt útil en investigacions policials. Disposant del graf social d'un grup delictiu es pot analitzar, per exemple, quin individu fa d'enllaç entre les diferents cèl·lules del grup. La detenció d'aquest individu pot suposar la desestructuració del grup sense haver hagut de realitzar detencions en cada una de les cèl·lules per aconseguir aquesta mateixa desestructuració. L'anàlisi de xarxes socials i, en concret, la correspondència inexacta de grafs ha estat utilitzat per estudiar les connexions entre els terroristes de l'11-S a [5].

- *Phishing:* La informació que conté el perfil d'un usuari pot ser molt valuosa per a un atacant que desitja fer-se passar per un conegut de l'usuari o per una empresa en la qual l'usuari confia. En el primer cas, l'atacant disposa dels noms dels amics i de la informació dels seus perfils, el que permet que pugui suplantar-los amb més facilitat. En el segon cas, l'atacant disposa de la informació personal que es troba al perfil de l'usuari i que pot contribuir en l'elaboració de correus electrònics més creïbles (per exemple, simulant que provenen de l'empresa per la qual treballa la víctima).

[1] Una definició més formal de que es considera un node ben comunicat es pot trobar al Capítol 4

CAPÍTOL 3

Recollida d'informació

En aquest capítol s'exposa el procés de recollida d'informació que permet obtenir els grafs socials amb els quals treballarem. El procés de recollida consta de quatre fases: selecció de les fonts, obtenció de la informació, emmagatzemament de la mateixa i representació dels grafs obtinguts, totes elles descrites en cada una de les quatre seccions del capítol. La primera secció conté una petita descripció de les xarxes socials *online* en les quals basarem el nostre anàlisi. La segona secció descriu de forma genèrica l'eina que es desenvoluparà per obtenir la informació de les diferents xarxes socials. La tercera secció inclou una petita descripció del sistema d'emmagatzemament utilitzat. La quarta i última secció conté una relació dels formats de representació de grafs que es faran servir.

3.1 Xarxes socials utilitzades

Avui en dia, el nombre de xarxes socials *online* és molt elevat i no para de créixer. Com que no totes elles podran ser analitzades, caldrà seleccionar-ne algunes en les quals centrar el nostre anàlisi. Les xarxes de les quals n'extraurem informació per complir els objectius d'aquest projecte són:

- *Twitter:* És un servei de *microblogging* que permet publicar missatges de fins a 140 caràcters. Els usuaris poden subscriure's a les actualitzacions que realitzen els altres usuaris, establint les relacions que permeten caracteritzar *Twitter* com a xarxa social. El graf social extret de *Twitter* és, per tant, un graf dirigit ja que les subscripcions no

són bidireccionals (l'Alice pot seguir en Bob, rebent totes les actualitzacions que aquest faci sense que necessàriament en Bob hagi de rebre els missatges que publiqui l'Alice). *Twitter* té 75 milions d'usuaris registrats.

- *Flickr:* És una comunitat virtual de compartició d'imatges i, des de fa poc, de vídeos. És un dels llocs més populars per emmagatzemar imatges, ja sigui per a compartir-les amb la comunitat o bé per encastar-les en altres pàgines web. Disposa de 32 milions d'usuaris registrats, el que la fa la xarxa més gran de les especialitzades en fotografia. Recentment *Flickr* ha superat els 4 bilions d'imatges emmagatzemades [15].

- *LastFM:* És un servei de recomanació de música. Disposa d'un sistema que analitza les preferències dels usuaris i en crea un perfil, que és visible a la pàgina de cada usuari. Els perfils dels usuaris es creen a partir de la informació de dues fonts: les emissores escoltades directament de *Lastfm* i les dades enviades per l'usuari de les cançons que escolta en el seu reproductor habitual (*Lastfm* ofereix uns *plugins* per als principals reproductors que permeten realitzar aquesta funció). La xarxa disposa de 30 milions d'usuaris actius.

- *Blogs:* Tot i que les bitàcores no són un servei de xarxa social *online* com les anteriors, el cert és que els *blogs* formen una xarxa social. Les relacions entre els creadors de *blogs* s'acostumen a explicitar en el *blogroll* on s'hi troben els *blogs* d'amics del creador o bé els que aquest considera interessants. Fent servir cada *blog* com a node i les relacions explicitades en el *blogroll* com a arestes, s'obtenen grafs socials que, com veurem posteriorment, presenten algunes característiques similars als grafs socials extrets de les altres OSNs.

- *TypePad:* És un servei de *blogs* dirigit a usuaris sense coneixements tècnics. A part de les funcions de publicació d'articles, inclou eines per a la creació d'àlbums fotogràfics. Algunes grans companyies britàniques com ara la *BBC* o *Sky News* fan servir *TypePad* per a elaborar els seus *blogs*.

3.2 Extracció de la informació

Com hem comentat anteriorment, la informació necessària per crear els diferents grafs socials i realitzar l'agregació dels mateixos és, en gran part, informació pública que es pot trobar a les pàgines de les diferents xarxes socials *online*. Un usuari podria extreure manualment aquesta informació, visitant amb un navegador web els perfils dels usuaris de cada OSN. Quan el nombre de xarxes socials analitzades o bé el nombre de nodes que es volen extreure és molt elevat, aleshores sorgeix la necessitat d'utilitzar programes que permetin automatitzar la tasca, agilitzant el procés de captura d'informació i fent viable una recollida a gran escala.

Els programes que accedeixen a Internet i n'extreuen qualsevol tipus d'informació se'ls coneix amb el nom genèric de *bot* (diminutiu de robot). Tot i així, hi ha altres termes més específics

per definir aquest tipus de programes: *spiders* (aranyes), agents i agregadors són tipus de *bots* especialitzats en alguna tasca concreta.

Les aranyes, també conegudes com a *crawlers* (rastrejadors), es caracteritzen per buscar altres pàgines a partir de la informació trobada en alguna pàgina coneguda. Les aranyes comencen la seva cerca en una o vàries pàgines inicials, que són escanejades en busca de referències a altres pàgines. Les referències que es van trobant seran també escanejades per l'aranya, continuant el procés de manera indefinida o bé fins que s'assoleix alguna condició de terminació. L'ús més representatiu de les aranyes és la indexació de pàgines web que realitzen els motors de cerca de la Web.

En el nostre cas, la recol·lecció d'informació per a la construcció dels diferents grafs serà feta per una aranya, que partint d'una adreça inicial (corresponent al perfil d'un usuari en una xarxa social concreta) buscarà relacions amb altres usuaris, que seran posteriorment visitats per la mateixa aranya.

3.2.1 Arquitectura bàsica d'un web-crawler

Tot i que els detalls d'implementació dels diferents *crawlers* que es fan servir habitualment són sovint ocultats pels seus creadors donada la importància que tenen actualment en el panorama econòmic, l'arquitectura a alt nivell d'un *web-crawler* és senzilla, tal i com es mostra a la Figura 3.1.

Un *crawler* estàndard està compost per un gestor de descàrregues (*downloader*), un analitzador (*parser*), una cua, un planificador i un dispositiu d'emmagatzemament. El *downloader* és el component que permet la interacció del *crawler* amb la xarxa a explorar. La seva tasca principal és la descàrrega de les pàgines que cal explorar per tal de buscar adreces a altres pàgines. Una vegada s'ha descarregat una pàgina, el contingut de la mateixa és analitzat pel *parser*. El *parser* n'extraurà la informació i la dipositarà al dispositiu d'emmagatzemament. També n'extraurà les adreces que hi hagi, que seran inserides a la cua per tal de ser processades posteriorment. Una vegada s'ha completat la descàrrega i anàlisi d'una pàgina, el planificador triarà la següent adreça a ser processada d'entre les adreces disponibles a la cua.

El procés descrit anteriorment pot ser infinit si es van trobant noves adreces a les pàgines que es van explorant. Per aquest motiu, s'acostuma a afegir alguna condició de terminació que permet acotar l'exploració o bé limitar l'exploració a un àmbit concret (per exemple, un mateix domini).

Una de les modificacions de l'arquitectura bàsica d'un *web-crawler* més habituals és l'ús d'un gestor de descàrregues multi-fil. Això permet augmentar la ràtio de descàrrega de les pàgines a explorar tot i que introdueix un *overhead* de paral·lelització que cal intentar minimitzar. Fer servir gestors de descàrrega multi-fil suposa un augment de l'ús de la xarxa, que pot arribar a ser, en molts casos, molt intensiu. És important considerar aquest fet, sobretot, quan les pàgines a explorar es trobin en un mateix servidor o bé requereixin l'ús d'un mateix

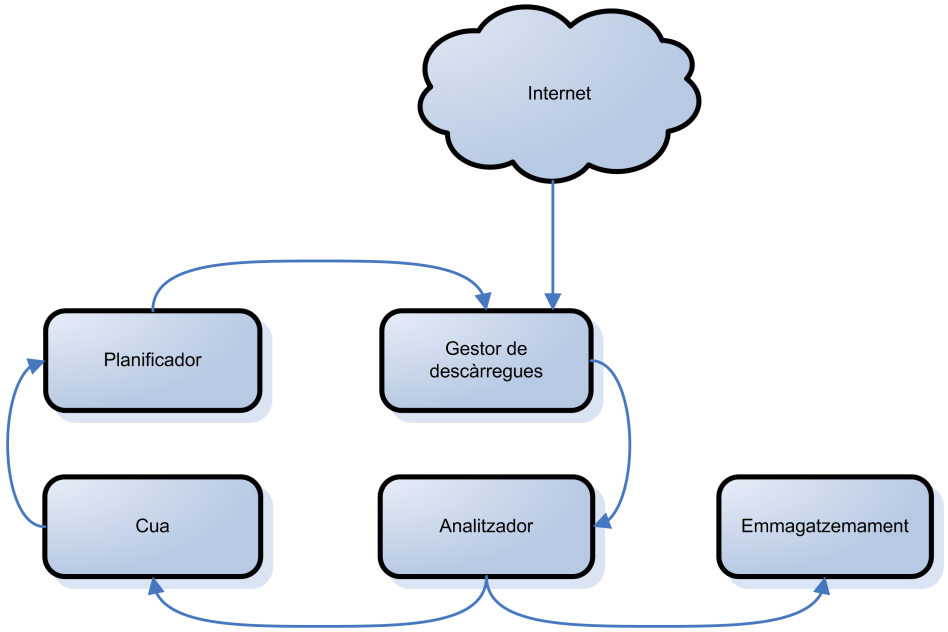

Figura 3.1: Arquitectura d'un *web-crawler*.

servei. En aquests casos i per tal de mantenir un comportament més educat a la xarxa, pot ser necessària la introducció expressa de temps d'espera entre peticions per tal de no saturar el servei explorat ni decrementar-ne notablement el seu rendiment. Una altra de les solucions que s'acostuma a implementar per intentar mitigar els efectes dels *crawlers* sobre el rendiment dels serveis és el Protocol d'Exclusió de Robots[28]. Aquest protocol proporciona un mecanisme als administradors per indicar quines parts de la web volen que siguin ignorades pels robots. Cal tenir en compte que els robots poden no obeir el protocol i que, per tant, aquest mètode no proporciona cap garantia que una part de la web no sigui analitzada pels robots que la visitin.

3.3 Emmagatzemament de la informació

La informació que el *crawler* va recollint ha de ser emmagatzemada per al seu posterior anàlisi. La informació recollida és, en essència, un graf: es recullen usuaris, informació dels usuaris i informació de les seves relacions que són, respectivament, nodes, atributs dels nodes i atributs de les arestes (i les arestes en sí mateixes). La quantitat d'informació que es pot arribar a recollir és molt elevada i, per tant, cal disposar d'una manera eficient per emmagatzemar-la mentre es va recol·lectant. En el nostre cas, les dades que es recullen durant el procés de *crawling* són dipositades en una base de dades relacional. Això permet anar escrivint els resultats sense haver de mantenir fitxers grans oberts i proporciona una gran versatilitat a l'hora de tractar amb les dades recollides. Un altre dels avantatges que ofereix treballar amb una base de dades relacional és poder anar accedint a les dades que

s'han anat recol·lectant, disposant de la informació actualitzada en tot moment. El control de la integritat i de duplicats queda delegat al gestor de la base de dades, el que ens ofereix la llibertat de no haver-ho de tractar manualment.

Una vegada s'ha acabat el procés de *crawling*, caldrà analitzar les dades obtingudes i, probablement, es voldrà visualitzar el graf generat. Arribat aquest punt, caldrà exportar les dades emmagatzemades a la base de dades a algun format reconeixible pel programari que es vulgui utilitzar.

3.4 Representació

Avui en dia, no existeix cap format estàndard per a la representació de grafs. La majoria de programes que treballen amb grafs fan servir un format propi per representar-los. Això fa que existeixin gran quantitat de formats diferents, cosa que suposa un problema a l'hora de treballar amb els mateixos grafs en diferents aplicacions. La creació d'un estàndard facilitaria notablement aquest procés. Si més no, existeixen alguns intents de crear aquest estàndard. Actualment, s'està considerant establir el *GXL* (*Graph eXchange Language*) com a format d'intercanvi entre els diferents formats de grafs, proporcionant un punt d'interoperabilitat entre les diferents eines de tractament de grafs.

A nivell pràctic, els formats més estesos per a la representació de grafs són *dot* i *GML*. Les dues alternatives van ser dissenyades com a format de representació de grafs per dos conjunts d'eines de tractaments de grafs (respectivament, *GraphViz* i *Graphlet*) i el seu ús s'ha anat estenent per la majoria de programari de tractament de grafs.

En general, tots els formats de representació es poden classificar en dues categories, depenent de si contenen o no informació de localització dels nodes, és a dir, el *layout* del graf. Així, hi ha un primer grup de formats que només contenen els nodes (i la informació associada a ells) i les arestes que els interconnecten (i la informació associada a les arestes). Aquests formats contenen tota la informació del graf, però no són útils a l'hora de visualitzar-lo. Per tal de poder visualitzar el graf, ens caldrà afegir informació sobre la localització dels nodes i les descripcions de les corbes que formaran les arestes. Els formats que contenen informació sobre el posicionament del graf formen el segon grup.

Per tal de visualitzar grafs que es trobin definits en qualsevol dels formats d'aquest segon grup, només caldrà utilitzar algun programari capaç de mostrar la informació continguda en els fitxers per pantalla. En canvi, si volem visualitzar un graf emmagatzemat amb algun dels formats del primer grup, caldrà realitzar càlculs auxiliars per determinar on situar cada node del graf i com dibuixar les diferents arestes.

CAPÍTOL 4

Anàlisi de la informació

Aquest capítol està dedicat a descriure la informació que podem extreure dels grafs socials. El capítol està dividit en dues seccions. La primera secció descriu algunes mesures aplicables als grafs que tenen especial importància en l'anàlisi de grafs socials. La segona secció descriu l'algorisme que es farà servir en el procés d'agregació d'informació.

4.1 Informació associada a un graf

La teoria de grafs defineix una gran quantitat de mesures i tècniques d'anàlisi de grafs que permeten estudiar-los amb precisió. Algunes d'aquestes mesures tenen un especial interès en l'anàlisi de les xarxes socials. En aquesta secció, es presenten les principals mesures utilitzades en aquest anàlisi de xarxes socials.

4.1.1 Diàmetre

El diàmetre d'un graf és la longitud del major dels camins més curts entre qualsevol parell de nodes del graf. El diàmetre d'un graf és, per tant, una mesura del major nombre de vèrtexs que es poden haver de travessar per anar d'un vèrtex del graf a qualsevol altre seguint el camí més curt.

L'experiment de Milgram [47] era un intent de mesurar el diàmetre del graf social dels Estats Units. Les xarxes socials són xarxes *small world*, caracteritzades per presentar un diàmetre petit. En una xarxa social, la distància entre dos nodes qualssevol és petita.

Definició El diàmetre efectiu o excentricitat és el mínim nombre de salts que permeten connectar una fracció (per exemple, el 90%) de tots els nodes del graf.

Per calcular el diàmetre efectiu d'un graf es pot fer servir el *hop-plot*:

Definició El *hop-plot* és la gràfica resultant de mostrar com evoluciona N_h en funció de h. El càlcul de N_h es realitza de la següent manera. Començant per un node u del graf, es calcula el nombre de veïns $N_h(u)$ que es troben a distància h o inferior. Es repeteix el procediment per a cada node del graf i se sumen els resultats per trobar la mida total del veïnatge per h salts ($N_h = \sum N_h(u)$).

4.1.2 Coeficient d'agrupament

El coeficient d'agrupament local és una mesura que s'aplica als nodes d'un graf i que permet quantificar com de ben connectats es troben els veïns d'un node entre ells. El coeficient d'agrupament és màxim (i per tant, val 1) quan tots els veïns d'un node es troben completament connectats, és a dir, formen un graf complet entre ells (Figura 4.1, graf B) i, en canvi, és mínim (val 0) quan no hi ha cap connexió entre els veïns del node analitzat (Figura 4.1, graf A). En concret, es defineix el coeficient d'agrupament com el nombre de connexions entre els veïns d'un node dividit pel nombre màxim de connexions que hi podria haver entre ells (si n és el nombre de veïns, $n * (n-1)/2$ per a grafs simètrics).

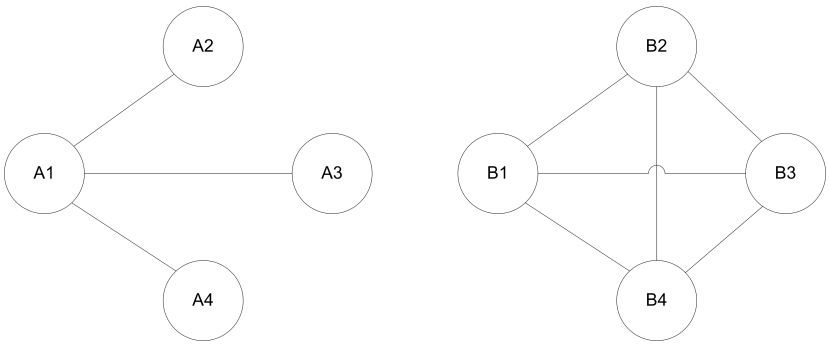

Figura 4.1: Coeficient d'agrupament.

El coeficient d'agrupament local d'un graf es defineix com la mitjana dels coeficients d'agrupament de tots els nodes del graf.

La majoria de xarxes obtingudes amb dades reals i, especialment, les xarxes socials, es caracteritzen per presentar un coeficient d'agrupament alt ([50], [25]), el que indica que els nodes s'acostumen a estructurar en petits grups fortament connectats entre ells i poc connectats a la resta de grups.

Un concepte que es fa servir sovint en SNA quan es parla d'agrupament és el de clique. Un clique és un subconjunt dels nodes del graf on tots els nodes del conjunt són adjacents entre

ells, és a dir, un subconjunt de nodes on el subgraf induït que formen és un graf complet. D'aquesta manera, podem dir que el coeficient d'agrupament local d'un node és màxim quan els seus veïns formen un clique.

4.1.3 Mesures de centralitat de nodes

Les mesures de centralitat són mètriques que s'apliquen als vèrtexs d'un graf i que en permeten determinar la seva importància relativa en el graf, seguint algun criteri concret. Són un dels conceptes més estudiats en l'anàlisi de xarxes ja que permeten analitzar una gran varietat de fenòmens i comportaments. Per exemple, les mesures de centralitat permeten estudiar el flux d'informació, la propagació d'infeccions o de rumors i el moviment de paquets, entre d'altres. A nivell pràctic, aquestes s'han utilitzat per a propòsits molt diversos al llarg dels anys. Un dels primers estudis en el que es van fer servir mesures de centralitat va ser realitzat per Cohn i Marriot l'any 1958 [16], en un intent d'explicar com era possible que es pogués administrar un país tan gran i amb una societat tan heterogènia com l'Índia. Un altre dels exemples d'aquest ús pràctic és la identificació i detenció de persones que es troben en punts clau d'una xarxa de delinqüència. És possible que, amb la detenció d'un sol individu que tingui un paper important dins d'una organització criminal, es neutralitzi l'efecte de tota l'organització, evitant que pugui ser operativa.

La centralitat és un concepte bastant intuïtiu i les mesures bàsiques de centralitat han estat un intent de formalitzar aquest concepte. S'assumeix universalment que, donat el graf de la Figura 4.2, el node 5 és estructuralment més central que qualsevol dels altres nodes del graf. És obvi que aquest node gaudeix d'una posició única en el graf: té el màxim grau, està el més proper possible a la resta de nodes del graf i es troba a la geodèsica entre tots els parell de punts.

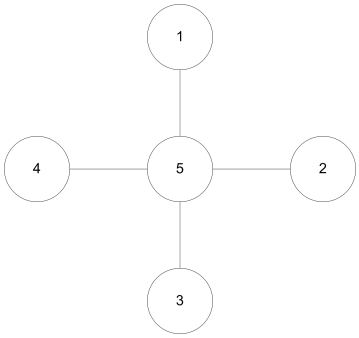

Figura 4.2: Graf estrella.

Dels diversos intents de determinar la manera de definir com n'és de central aquesta posició n'han sorgit les tres mesures bàsiques de centralitat: la centralitat de grau (*degree centrality*), la centralitat intermèdia (*betweenness centrality*) i la centralitat de proximitat (*closeness centrality*).

La primera definició de centralitat es basa en la relació directa entre la centralitat i el grau d'un node.

Definició La centralitat de grau (*degree centrality*) defineix la centralitat d'un node com el grau d'aquest.

$$C_D(v_0) = grau(v_0) \tag{4.1}$$

La centralitat de grau és la mesura de centralitat més simple i més intuïtiva. Va ser utilitzada per primer cop per Shaw [42] al 1954 i utilitzada posteriorment per molts altres autors ([31], [19], [36], [39]), els quals van desenvolupar tot un conjunt de mesures de centralitat basades en el grau d'un node. Si parlem, per exemple, de comunicacions en una xarxa social, un node amb un grau elevat es troba connectat a un gran nombre de nodes, fent que aquest es trobi en "més converses". Podem comprovar com, en el graf de l'exemple, el node 5 té la centralitat de grau més elevada del graf.

El segon enfoc de la centralitat es basa en la freqüència en la que un punt es troba entre un altre parell de punts en la geodèsica que els interconnecta. A la Figura 4.3 podem observar les deu geodèsiques del graf anterior. El punt 5 es troba enmig de sis d'elles (a les altres quatre n'és un extrem). Per tant, el punt 5 presenta la centralitat intermèdia més alta de tot el graf. Parlant una vegada més de les comunicacions d'una xarxa social, una persona que es troba estratègicament situada en les línies de comunicació que lliguen a altres parells de persones direm que és central ([42], [6]). Una persona situada en aquest punt pot influir en un grup ocultant o distorsionant la informació que transmet. Aquestes persones tenen responsabilitat en la conservació de la comunicació ([43]) i són potencialment coordinadores de processos grupals ([16]).

Quantificar la intermediació és senzill quan per cada parell de nodes del graf només hi ha una geodèsica (com a l'exemple anterior). En canvi, quan hi ha vàries geodèsiques que connecten dos nodes (per exemple, el graf de la Figura 4.4), la quantificació és més complexa.

Definició La centralitat intermèdia (*betweenness centrality*) d'un node es defineix com a

$$C_B(v_0) = \sum_{\substack{i,j=1 \\ i \neq j}}^{n-1} \frac{\sigma_{v_i v_j}(v_0)}{\sigma_{v_i v_j}} \tag{4.2}$$

on $\sigma_{v_i v_j}$ és el nombre de camins de longitud mínima entre v_i i v_j i $\sigma_{v_i v_j}(v_0)$ és el nombre de camins de longitud mínima entre v_i i v_j que passen per v_0.

El tercer enfoc de la definició de centralitat té en compte com de proper es troba un node a la resta de nodes del graf. Tornant a l'exemple inicial de la Figura 4.2, el punt 5 es troba a distància 1 de tots els altres nodes del graf mentre que cada un dels altres nodes es troba a

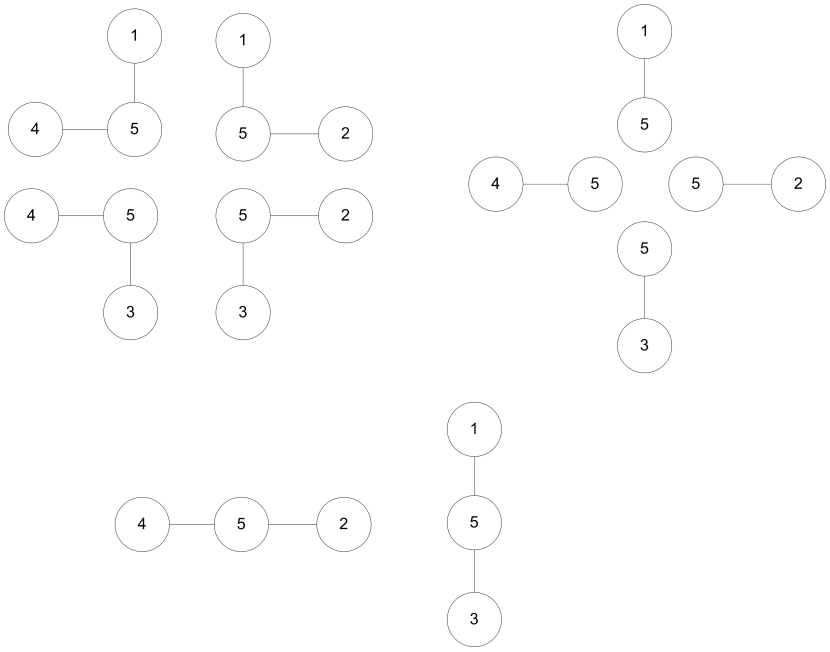

Figura 4.3: Geodèsiques del graf estrella.

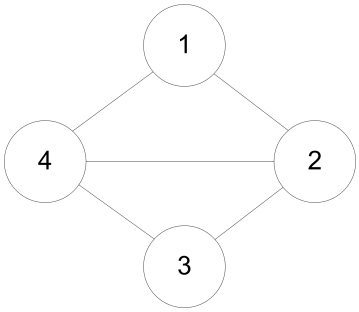

Figura 4.4: Graf amb més d'una geodèsica per parell de nodes.

distància 1 del node 5 i a distància 2 de la resta de punts. Seguint aquest enfoc, el node 5 és, de nou, el node més central. Aquest tercer enfoc també està relacionat amb el control de la comunicació tot i que de manera diferent. Ara, un punt es considera central en la mesura que pot evitar el potencial d'altres per a controlar la comunicació. Bavelas ([7]) deia que una posició no era central quan "necessita transmetre missatges a través d'altres". D'aquesta manera, una posició és central quan no depèn d'altres com a intermediaris, és a dir, quan és independent. La independència d'un node es troba determinada per la seva proximitat a tota la resta de nodes del graf. Altres autors ([24], [40]) van generalitzar aquesta idea al definir el node més central d'una xarxa de comunicacions com aquell per al qual el cost de la comunicació amb els altres nodes és mínim. La mesura de centralitat de proximitat d'un node més simple es defineix sumant les distàncies geodèsiques d'aquell node a la resta de nodes del graf. De fet, aquesta és una mesura de centralitat inversa ja que la mesura creix quan els nodes es troben més allunyats.

Definició La centralitat de proximitat (*closeness centrality*) d'un node és l'invers del sumatori de les distàncies geodèsiques a tots els altres nodes del graf.

$$C_C(v_0) = \frac{1}{\sum_{i=1}^{n-1} d(v_0, v_i)} \tag{4.3}$$

En resum, podem dir que la centralitat d'un node d'un graf pot venir determinada per tres factors diferents: el grau, la intermediació i la proximitat. L'elecció de la mesura de centralitat a utilitzar dependrà del context concret del cas que s'estigui estudiant. L'interès per l'activitat comunicativa apuntarà cap a una mesura basada en el grau, l'interès pel control de la comunicació en una mesura basada en la intermediació i l'interès per la independència en una mesura basada en la proximitat.

Mesures de centralitat ponderades

La magnitud d'una mesura de centralitat de grau depèn, en gran part, de la mida del graf que s'estigui analitzant. És evident que, en un graf format per quatre nodes, un node de grau tres és un node amb una centralitat molt alta mentre que, en un graf format per un milió de nodes, un node de grau tres tindrà, probablement, una centralitat molt baixa. La definició bàsica de centralitat de grau és útil per a quantificar la centralitat d'un node respecte a altres nodes del mateix graf però no ho és a l'hora de comparar-la amb altres grafs. Amb la idea de resoldre aquest problema, sorgeix una segona mesura de centralitat de grau que compensa les diferències entre grafs de diferents mides. Tenint en compte que donat un node $v \in V$, si $|V| = n$ aleshores v pot tenir com a molt $n-1$ veïns, es defineix la centralitat de grau ponderada.

Definició La centralitat de grau ponderada d'un node v ($C'_D(v)$) és el grau del node entre el nombre màxim de veïns que podria tenir.

$$C'_D(v) = \frac{C_D(v)}{n-1} = \frac{grau(v)}{n-1} \tag{4.4}$$

De la mateixa manera que la centralitat de grau s'havia de compensar per tal de poder fer-la servir per comparar nodes en diferents grafs, la centralitat intermèdia necessita també aquesta normalització. El nombre màxim de parells de nodes de V sense tenir en compte el node v que s'està avaluant és $(n-1)(n-2)$, per tant:

Definició La centralitat intermèdia ponderada d'un node v ($C'_B(v)$) és la centralitat intermèdia entre el nombre màxim de parells de nodes excepte el node avaluat.

$$C'_B(v) = \frac{C_B(v)}{(n-1)(n-2)} \tag{4.5}$$

Per últim, podem observar com la centralitat de proximitat també depèn de la mida del graf. En aquest cas i atès que la fórmula està basada en les distàncies entre un node v i els altres $n-1$ nodes, la normalització de la centralitat de proximitat es realitzarà dividint per $n-1$.

Definició La centralitat de proximitat ponderada d'un node v ($C'_C(v)$) correspon a la centralitat de proximitat entre el nombre de nodes del graf menys 1.

$$C'_C(v) = \frac{C_C(v)}{n-1} \tag{4.6}$$

4.1.4 Detecció de comunitats

Com hem vist anteriorment, el coeficient d'agrupament permet mesurar com de ben connectats es troben entre ells els veïns d'un node. De manera similar, podem definir com de ben connectats es troben un conjunt de nodes qualsevol del graf. Una de les propietats que presenten les xarxes socials és l'estructuració en conjunts de nodes fortament relacionats entre ells i poc relacionats amb la resta de nodes del graf, és a dir, l'estructuració en comunitats. La detecció i identificació d'aquestes comunitats permet facilitar la visualització i la comprensió de la xarxa analitzada.

Per tal de detectar i definir aquestes comunitats, es fan servir principalment dos tipus d'algorismes: aglomeratius i divisius. Els algorismes aglomeratius parteixen del graf a analitzar sense cap aresta i van afegint arestes segons alguna mesura de similitud definida entre els nodes. D'aquesta manera, les comunitats queden definides entre els nodes que es troben en un mateix component connex. En canvi, els algorismes divisius realitzen el procés invers: parteixen del graf a analitzar original (amb totes les arestes) i van eliminant arestes. Les arestes eliminades es poden seleccionar tenint en compte la similaritat dels nodes que uneixen (en aquest cas, s'eliminen les que uneixen nodes menys similars) o bé tenint en compte la centralitat de les arestes, una mesura anàloga a les mesures de centralitat de nodes explicades a la Secció 4.1.3.

Tots dos tipus d'algorismes són iteratius: a cada pas, es calcula la mesura a fer servir, se selecciona l'aresta a afegir o eliminar, s'afegeix o elimina l'aresta i es torna a començar. Aquest procés presenta diversos problemes que cal tractar. En primer lloc, el fet de tornar a calcular a cada pas la mesura per tot el graf suposa un cost computacional alt. Per aquest motiu, alguns algorismes ometen aquest pas i realitzen totes les decisions amb els càlculs realitzats a la primera iteració. Els resultats obtinguts amb aquesta simplificació seran, per tant, una mica menys precisos, motiu pel qual caldrà decidir si es prioritza la precisió o bé el temps de càlcul. En segon lloc, és necessari establir un criteri d'aturada de l'algorisme. Depenent de l'objectiu que es persegueixi, el criteri d'aturada pot ser el nombre de comunitats a crear, la mida d'aquestes comunitats o el moment en el qual les comunitats detectades siguin les més representatives del graf. En aquest últim cas, es defineixen mesures que permeten quantificar com de representativa és una partició en comunitats.

Al graf de la Figura 4.5 es poden observar, a primer cop d'ull, dues comunitats ben diferenciades: una formada pels nodes de l'1 al 6 i l'altre formada pels nodes del 7 al 12. Una possible partició d'aquest graf seria, per tant, en aquestes dues comunitats de 6 nodes cada una. Però aquesta partició no és única ja que dins de cada una de les comunitats es podria realitzar una segona divisió, aconseguint en total quatre comunitats de tres nodes cada una (1-2-6, 3-4-5, 7-8-12 i 9-10-11). La selecció d'una o altra partició del graf dependrà, per tant, de la condició de terminació de l'algorisme que es faci servir.

L'evolució de les comunitats detectades a cada iteració de l'algorisme es representa, habitualment, en forma de dendrograma. Un dendrograma és una representació gràfica en forma d'arbre on es pot veure l'evolució de les agrupacions dels nodes conforme es va executant l'algorisme. Realitzant un tall horitzontal al dendrograma, obtenim les comunitats detectades en un moment concret de l'execució. A la Figura 4.5 hi trobem representat el dendrograma corresponent al graf que l'acompanya i s'hi poden apreciar les dues possibles particions comentades anteriorment en dos nivells diferents del dendrograma.

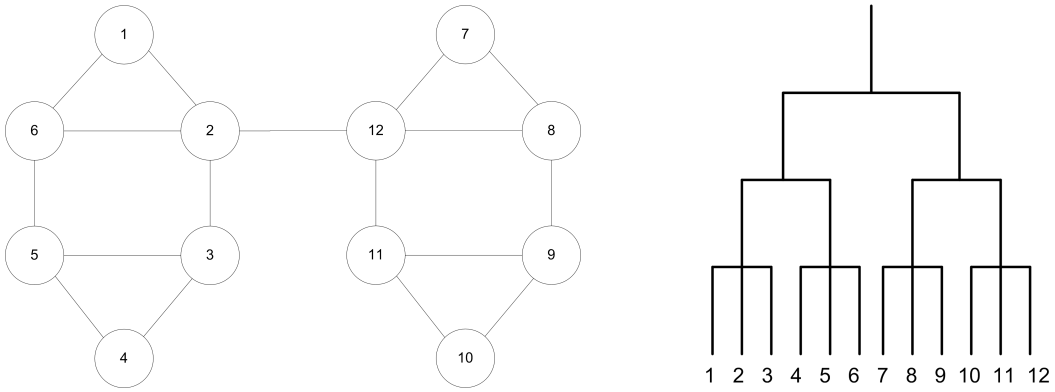

Figura 4.5: Detecció de comunitats: graf i el seu corresponent dendrograma.

4.2 Tècniques d'agregació de la informació

Tal com hem descrit al Capítol 1, l'objectiu d'aquest projecte és l'obtenció del graf social d'una persona o comunitat utilitzant agregació d'informació pública de diferents fonts. Per tal d'agregar informació de diferents fonts caldrà trobar les correspondències entre els nodes dels diferents grafs obtinguts (identificar persones en els diferents grafs). Com que els grafs obtinguts de les diferents fonts no seran complets i que, per tant, hi haurà individus que apareixeran en algun dels grafs i no ho faran en altres, i que les relacions expressades en cada una de les xarxes socials poden ser diferents, no serà possible trobar una correspondència exacta entre els diferents grafs. Per aquest motiu, caldrà buscar la solució que més s'aproximi. En aquest cas, parlarem de correspondència de grafs inexacta.

El problema de la correspondència de grafs inexacta és NP-complet [1]. S'han plantejat diferents tipus d'algorismes per solucionar-lo, que van des de la utilització de conjunts difusos

fins a l'ús d'algorismes genètics, passant per mètodes probabilístics, xarxes neurals, tècniques d'agrupament...

Com hem vist a la Secció 2.2.3, en els últims anys s'han presentat nombrosos articles que fan ús de la correspondència inexacta de grafs per tal de realitzar atacs de desanonimització contra xarxes socials. Aquest tipus d'atacs se centren en trobar la correspondència entre dos grafs, un dels quals ha passat per un procés d'anonimització on s'ha eliminat part de la informació que conté (normalment, s'eliminen part dels atributs dels nodes i de les arestes) i s'hi ha afegit soroll (s'eliminen i es creen nodes i arestes). La desanonimització de grafs és un problema similar a l'agregació d'informació de diferents grafs i, per tant, els algorismes que es fan servir per tractar el primer problema poden ser utilitzats per a solucionar el segon.

Els processos de desanonimització de grafs es basen en la reidentificació dels nodes i es porten a terme a partir d'informació auxiliar obtinguda de fonts externes. Així, s'acostuma a partir d'un graf G que ha passat per un procés d'anonimització i d'un altre graf G_{aux} que s'ha obtingut amb informació pública o bé d'alguna font diferent de la que ha originat el graf G, i s'intenta trobar correspondències entre els dos grafs amb l'objectiu de reidentificar els nodes de G. Aquest procés presenta una gran similitud amb les tècniques d'agregació d'informació de grafs que es tracten en aquest projecte. Tot i que la nomenclatura és diferent (ara partim de dos grafs G_A i G_B que no necessàriament han d'haver estat anonimitzats), el procés per arribar a l'agregació és el mateix: cal identificar correspondències en els dos grafs per tal de millorar i completar la informació que es disposa de cada usuari.

Existeixen diversos algorismes que permeten trobar correspondències entre nodes de diferents grafs, establint la base del procés d'agregació. Dels diferents algorismes comentats a la Secció 2.2.3, ens centrarem en l'algorisme proposat al març de 2009 per Narayanan i Shmatikov a [35] ja que permet realitzar l'agregació de grafs a gran escala de manera passiva (no requereix haver de modificar la xarxa analitzada).

L'algorisme pren com a entrades dos grafs ($G_A = (V_A, E_A)$ i $G_B = (V_B, E_B)$) i un conjunt de correspondències inicials ($C_0 \in V_A \times V_B$) entre els nodes d'aquests que servirà com a llavor per inicialitzar l'algorisme. A l'acabar la seva execució, l'algorisme retorna un altre conjunt amb les correspondències finals. El conjunt de correspondències finals, serà sempre una ampliació de l'inicial o, en el pitjor dels casos, el mateix conjunt inicial.

El procés d'agregació es divideix, per tant, en dues fases: inicialització i propagació. La fase d'inicialització consisteix en determinar les correspondències inicials entre els nodes que serviran de llavor per a la fase de propagació. La fase de propagació consistirà en buscar noves correspondències a partir de les llavors donades.

Hi ha diversos mètodes que permeten buscar les correspondències inicials que serviran de base per iniciar l'agregació, entre els quals trobem els atacs actius comentats a la Secció 2.2.3. En el nostre cas, s'aprofitarà el nom d'usuari dels nodes de les xarxes a analitzar per tal de simplificar al màxim aquesta fase d'inicialització. Les correspondències inicials seran determinades buscant noms d'usuari coincidents a les dues xarxes.

Una vegada seleccionades les correspondències inicials, es pot començar la fase de propagació que, com el seu nom indica, consistirà en propagar aquestes correspondències per la resta dels grafs fent servir l'estructura topològica de les xarxes a agregar. A cada iteració, l'algorisme comença amb un conjunt de correspondències inicials (per la primera iteració, seran el resultat de la fase d'inicialització). S'elegeix un node qualsevol $a \in V_A$ que encara no tingui una correspondència associada i es calcula una puntuació per cada node $b \in V_B$ que encara no tingui correspondència. La puntuació donada és igual al nombre de veïns d'a que ja tenen una correspondència associada amb veïns de b.

Les Figures 4.6 i 4.7 mostren dos exemples de com funciona aquest procés de càlcul de la puntuació dels diferents nodes per dos grafs A i B. A la Figura 4.6, es mostra el cas més simple: el conjunt inicial de correspondències és $\{(A1, B1)\}$ i s'ha seleccionat el node $A2$ per buscar-ne de noves. El node $A2$ només té un veí que es trobi a la llista de correspondències: $A1$. Per tant, els veïns de $B1$ (corresponent a $A1$ segons la llista inicial) que no tinguin cap correspondència associada incrementaran la seva puntuació. El mateix procés però amb un nombre de nodes més elevat pot ser observat a la Figura 4.7. En aquest cas, el conjunt de correspondències inicial és $\{(A2, B3), (A3, B2), (A4, B5)\}$ i el node inicial és $A1$. El node $A1$ disposa ara de 3 veïns que es trobin a la llista de correspondències ($A2$, $A3$ i $A4$). Per tant, els veïns de $B3$, $B2$ i $B5$ que no tinguin cap correspondència associada incrementaran la seva puntuació. Tant $B2$ com $B5$ només tenen un veí sense correspondència, $B1$, que incrementa la seva puntuació en dos punts, un per cada un dels nodes. El node $B3$ té dos veïns sense correspondència, $B6$ i $B1$, que incrementen en un punt la seva puntuació. D'aquesta manera, el node $B1$ és seleccionat com a corresponent a A1 ja que obté la puntuació més alta amb 3 punts.

Figura 4.6: Agregació: cas simple.

Una vegada obtingudes les puntuacions dels nodes de V_B, l'algorisme mesura si són suficients per establir una nova correspondència. Per fer-ho, es defineix l'excentricitat, que mesura com es destaca un element concret de la resta d'elements. L'algorisme establirà una correspondència parcial entre el node a i el node amb la puntuació més alta de V_B si l'excentricitat

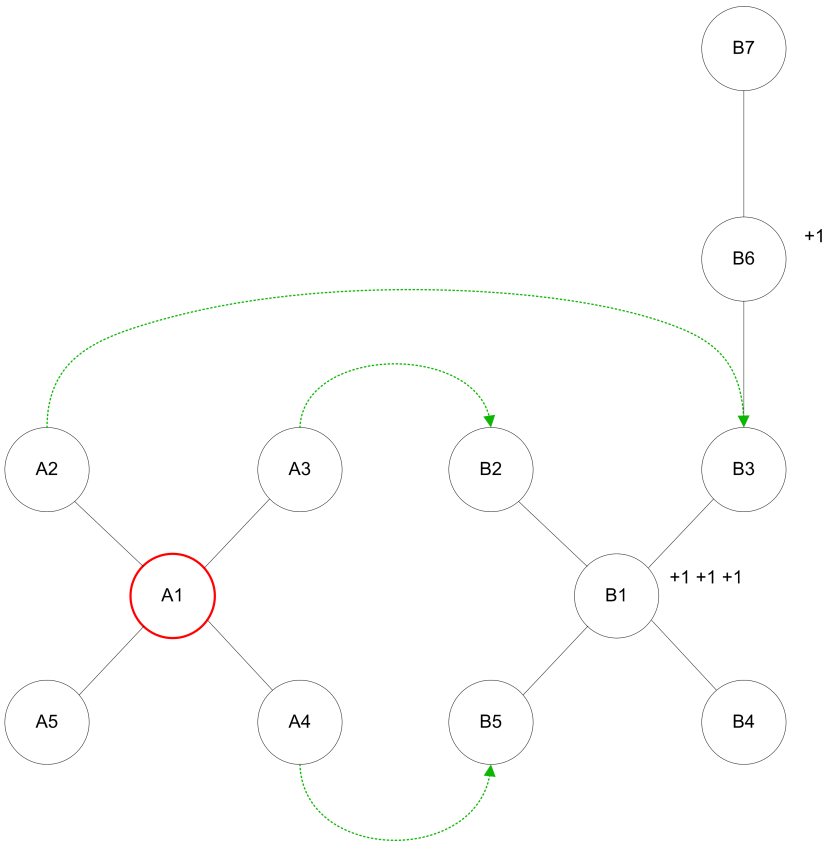

Figura 4.7: Agregació: segon exemple.

de les puntuacions obtingudes supera un valor llindar α. En cas contrari, s'entén que les puntuacions són massa similars entre elles com per prendre la decisió i, per tant, no s'estableix la correspondència. El cas mostrat a la Figura 4.8 és un exemple en el qual l'algorisme no serà capaç de decidir la correspondència del node $A2$ ja que tant el node $B2$ com el $B4$ obtenen la mateixa puntuació. Una vegada obtinguda la correspondència parcial cal comprovar-la abans de donar-la com a vàlida. Com que l'algorisme no fa diferències entre quin dels dos grafs analitzats és utilitzat com a G_A i quin com a G_B, es realitza el mateix procés però intercanviant els grafs i calculant les puntuacions per al node de V_B. Si el resultat coincideix, aleshores la correspondència es dóna per vàlida i és afegida a la llista de correspondències.

Les puntuacions definides anteriorment es troben esbiaixades a favor dels nodes amb un grau elevat. Per compensar aquest biaix, la puntuació de cada node es divideix per l'arrel quadrada del seu grau.

L'algorisme descrit no defineix com se seleccionen els nodes a analitzar en cada moment ni especifica explícitament quins criteris cal fer servir per determinar quantes vegades es visita un node en busca de la seva correspondència ni quan s'atura l'algorisme. La implementació realitzada resol aquestes qüestions definint fases. A cada fase, tots els nodes que no tenen una correspondència associada són analitzats. L'algorisme executa com a màxim un número prefixat de fases que rep com a paràmetre de configuració. També a cada fase, el valor

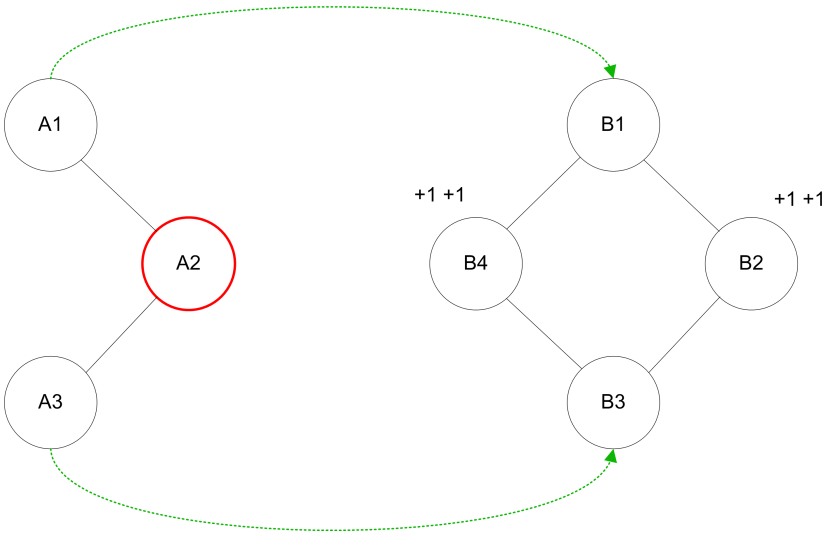

Figura 4.8: Agregació: cas amb empat

llindar α és multiplicat per un factor αDec que el decrementa. Això permet que cada nova fase sigui menys exigent a l'hora d'establir correspondències, augmentant el nombre total de correspondències trobades. Valors petits d'αDec donen com a resultat més correspondències mentre que valors grans (propers a 1) d'aquest paràmetre produeixen menys correspondències però més fiables. L'algorisme va executant fases fins que, o bé arriba al número màxim definit, o bé les correspondències donades a l'última fase són iguals a les de la fase anterior.

CAPÍTOL 5

Aplicacions

Aquest capítol està dedicat a les diferents aplicacions utilitzades i desenvolupades durant la realització d'aquest projecte. El capítol es troba dividit en tres parts, cada una de les quals inclou les aplicacions utilitzades amb una mateixa finalitat. La primera secció està dedicada a les eines de recollida d'informació. La segona secció explica les eines de visualització d'informació que s'han fet servir. Per últim, la tercera secció se centra en les eines utilitzades per analitzar els grafs obtinguts.

Per qüestions d'espai, aquesta memòria no inclou explicacions detallades de les aplicacions desenvolupades (només es proporciona, a tall d'exemple, el codi que implementa l'algorisme d'agregació). Tot i així, la memòria va acompanyada d'un CD on s'hi pot trobar el codi de les aplicacions desenvolupades en aquest projecte.

5.1 Recollida d'informació: l'aplicació de web-crawling

Per tal de realitzar la recollida de dades s'ha desenvolupat un *web-crawler* capaç d'explorar les diferents OSN i extreure'n la informació necessària per construir el graf social de cada una d'elles. L'aplicació ha estat desenvolupada fent servir *Java*[17] com a llenguatge de programació i *MySql*[18] com a sistema gestor de la base de dades. L'elecció d'aquestes eines va ser realitzada en base a tres criteris: velocitat de desenvolupament, execució multiplataforma i llicència GPL.

5.1.1 El gestor de descàrregues

El gestor de descàrregues és el mòdul del *crawler* encarregat d'interactuar amb l'OSN, descarregant-ne les pàgines amb els perfils dels usuaris a explorar. El gestor de descàrregues rep una URL del planificador i descarrega el seu contingut, que servirà d'entrada al *parser*.

El gestor de descàrregues implementat ofereix l'opció de configurar un temps d'espera entre la descàrrega d'una pàgina i la descàrrega de la següent. Tot i que fer servir un temps d'espera superior a 0 fa que el temps necessari per al *crawling* s'incrementi, això també permet ser més respectuosos amb la xarxa analitzada, evitant enviar múltiples peticions de manera continuada.

Una altra de les funcionalitats del gestor de descàrregues és la redirecció d'aquestes a través de la xarxa *Tor*[20]. Tor és una xarxa distribuïda de repetidors dissenyada per anonimitzar aplicacions basades en TCP com ara la navegació web. Per tal d'establir una connexió, el client crea un circuit a través de la xarxa *Tor* on cada un dels nodes només coneix el seu successor i el seu antecessor.

Per tal de poder redirigir les descàrregues a través de la xarxa *Tor*, cal tenir instal·lat el client de *Tor* a la màquina on s'executa el *crawler*. Quan s'activa aquesta funció, el *crawler* fa servir la llibreria *TorLib*[22] per redirigir totes les descàrregues a un port (9050 per defecte) de la màquina local on el client de *Tor* es troba escoltant . El propi client de *Tor* s'encarrega aleshores de crear els circuits necessaris i de la transmissió de les dades a través de la xarxa.

5.1.2 Els parsers

El *parser* és el component del *crawler* encarregat de buscar dins d'un perfil d'un usuari els enllaços als perfils dels usuaris amb els quals es troba relacionat. Aquesta tasca seria trivial si la Web semàntica fos avui una realitat, ja que totes les dades dels usuaris i les seves relacions es trobarien descrites de manera formal amb una mateixa ontologia. En concret, FOAF (*Friend of a Friend*) és l'ontologia utilitzada per descriure les persones i les seves relacions. Tot i que l'ontologia es troba definida des de fa força temps (la versió 1.1 de l'especificació data de l'abril de 2005[13]), de les OSN analitzades en aquest projecte, només una (*Twitter*) està reconeguda com a font de dades FOAF pel W3C [48]. Per aquest motiu, en comptes d'elaborar un sol *parser* capaç d'analitzar les dades en format FOAF, s'ha realitzat un *parser* per cada una de les OSN a analitzar, de manera que cada un dels *parsers* és capaç d'analitzar les dades obtingudes de cada OSN en el seu format propi.

Les dades dels usuaris i les seves relacions s'han obtingut de les pròpies OSN a través de dues vies: les pàgines dels perfils dels usuaris i les APIs de cada OSN. L'accés a les dades a través de les APIs és la manera més ràpida d'obtenir-les ja que aquestes inclouen funcions que retornen directament la informació buscada, normalment, en format XML. Algunes OSN no ofereixen obertament aquest servei, ja sigui perquè no disposen d'ell o bé perquè requereixen

estar donat d'alta com a usuari o desenvolupador per fer-lo servir. En aquests casos, es fa servir la segona via d'obtenció de dades: els perfils dels usuaris. Els perfils són més costosos d'analitzar ja que la informació buscada es troba en un document HTML pensat per a que un usuari el visualitzi amb un navegador i no per ser analitzat de manera automatitzada. A més, també suposen un augment en el temps de descàrrega de la informació ja que inclouen moltes més dades que després hauran de ser descartades. Tot i aquests inconvenients, aquest mètode és el més utilitzat per a explorar OSN en aquest projecte ja que de les OSN analitzades, només *Twitter* ofereix les dades de la seva API obertament[1].

El *parser* per a *Twitter* és, per tant, el més senzill de tots. El document que el *parser* rep del gestor de descàrregues és un XML amb un element[2] pare que conté tants fills com amics té l'usuari analitzat. Cada fill conté un sol element amb l'etiqueta *id*, el contingut del qual és directament l'identificador de l'usuari.

Els *parsers* per a *Flickr* i *Typepad* són també molt senzills de construir. En tots dos casos, només cal buscar dins la pàgina del perfil els enllaços que especifiquin una relació de tipus "contact". Aquests enllaços corresponen a les pàgines dels perfils dels contactes de l'usuari analitzat. Tot i que el *parser* és senzill, en aquest cas el fet de buscar els contactes d'un usuari ja suposa haver de processar tota una pàgina HTML, augmentant, per tant, el cost computacional de l'exploració de cada usuari. També en aquests casos cal tenir en compte que obtenir tota la llista de contactes d'un sol usuari pot suposar haver de descarregar i processar diverses pàgines ja que, si el nombre de contactes és elevat, aquests no es mostren mai en una sola pàgina. Aquest fet també suposa un augment del cost computacional d'analitzar un usuari de *Flickr* o *Typepad* davant d'un usuari de *Twitter*.

El *parser* per a *LastFm* és similar als dos anteriors però afegeix una altra comprovació. Ara, buscarem tots els enllaços que tinguin una estructura concreta (enllaços relatius que comencin amb */music/*) i que es trobin dins d'un *div* de classe *userContainer*. Per fer-ho, aquest *parser* actua com a una petita màquina d'estats amb només dos estats. Quan el *parser* es troba a l'estat inicial, no s'analitzen els enllaços trobats. Quan el *parser* es troba a l'estat final, s'analitzen els enllaços, emmagatzemant aquells que segueixin l'estructura anteriorment especificada. Les transicions entre els estats es produeixen quan s'entra o se surt del *div* de classe *userContainer*.

A diferència dels anteriors *parsers*, el *parser* de *blogs* és bastant complex. En aquest cas, no hi ha una estructura fixada on buscar els enllaços que corresponen als amics de l'usuari analitzat ni aquests enllaços tenen una forma concreta que els diferenciï de la resta d'enllaços. Això fa que no sigui possible extreure totes les relacions d'un usuari sense cometre cap tipus

[1] De fet, una gran part de la API de Twitter es troba restringida i només és accessible pels seus usuaris i desenvolupadors. Les funcions que permeten obtenir la informació sobre les relacions dels usuaris i la informació dels perfils són obertes i, per tant, s'han pogut fet servir per a extreure la informació necessària.

[2] En XML, s'anomena element a un component que comença amb una etiqueta inicial i finalitza amb una etiqueta final. El que hi ha entre l'etiqueta inicial i la final s'anomena contingut. El contingut d'un element pot ser tant text com altres elements.

d'error. Per aquest motiu, el *parser* per a *blogs* disposa de dos modes de funcionament, el mode conservador i el mode heurístic, que prioritzen, respectivament, la qualitat dels enllaços trobats i el nombre d'enllaços. El mode conservador funciona de manera similar als altres *parsers*, detectant quina plataforma s'ha fet servir per realitzar el *blog* i identificant-ne l'estructura concreta del *blogroll*. Aquest mode té l'avantatge de no proporcionar enllaços incorrectes (una vegada detectada la plataforma, l'estructura del *blogroll* és fixa). En canvi, presenta l'inconvenient de tenir limitats els *blogs* analitzables als que utilitzen plataformes identificables pel *parser*. Per aquest motiu, el mode de funcionament és anomenat conservador. El segon mode de funcionament, el mode heurístic, representa un canvi de prioritats. En aquest cas, el nombre d'enllaços trobats serà més elevat ja que es podrà analitzar qualsevol *blog* però és possible que alguns d'aquests enllaços siguin erronis, és a dir, que no representin enllaços a les pàgines dels amics de l'autor del *blog* analitzat. Aquest mode fa servir una funció heurística per determinar el *blogroll* del *blog* analitzat.

5.1.3 El planificador

El planificador és de vital importància per al funcionament del *crawler*. La tasca del planificador és seleccionar el proper node a ser explorat d'entre tots els nodes de la llista de pendents. Una elecció encertada permetrà augmentar la velocitat de descobriment dels nodes ocults de la xarxa. Per exemple, en el graf mostrat a la Figura 5.1, on el node 1 ja ha estat explorat i cal decidir si explorar el node 2 o bé el 3, és evident que elegir el node 2 ens permetrà descobrir altres nodes de la xarxa molt més ràpidament que si s'escull el 3. Havent explorat 3 nodes (1, 2 i 4) descobrim 10 nodes del graf, el que representa el 83% del total de nodes. En canvi, si el següent node a explorar després de l'1 hagués estat el 3, hauríem descobert només 4 nodes de tot el graf, el 33% de la totalitat de nodes.

Un altre dels factors a considerar a l'hora d'escollir l'algorisme que haurà d'implementar el planificador és el biaix que s'introdueix indirectament a les dades recollides. Tornant a l'exemple de la Figura 5.1 i havent explorat 3 nodes, podem veure com el grau mitjà del graf obtingut és molt diferent per als dos casos comentats. Si el node explorat en segon lloc és el 2, el grau mitjà del graf obtingut després d'haver explorat 3 nodes és 4.3 (els nodes 1,2 i 4 tenen, respectivament, graus 2, 5 i 6) , mentre que, si el node explorat en segon lloc és el 3, el grau mitjà serà 1.6 (els nodes 1, 3 i 8 tenen, respectivament, graus 2, 2 i 1).

Els diferents algorismes de planificació afecten, per tant, el resultat de les mesures que s'apliquin al graf, introduint un biaix que cal tenir en compte a l'hora de valorar els resultats. Hi ha diversos estudis que analitzen l'efecte dels mètodes de *crawling* utilitzats en l'obtenció de grafs sobre algunes de les mesures més utilitzades en SNA. A [51] s'analitza l'efecte dels algorismes de planificació sobre la velocitat de descobriment de nodes i enllaços en el graf, el grau mitjà i el coeficient d'agrupament. Els efectes sobre el grau mitjà, la centralitat intermèdia, la mida mitjana dels camins, l'*assortativity* i el coeficient d'agrupament són analitzats a [29]. Altres estudis que tracten el tema són [30] i [33].

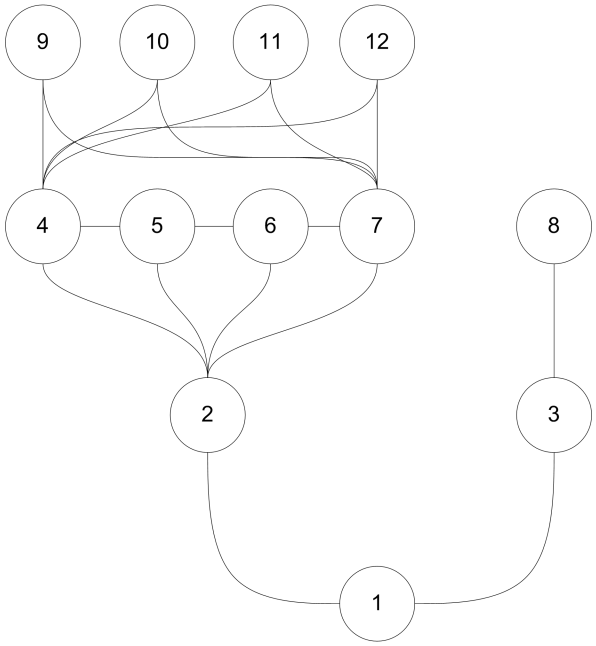

Figura 5.1: Graf a explorar.

Per tal de poder analitzar els efectes que els diferents algorismes de planificació tenen sobre els grafs obtinguts, el *crawler* implementat disposa de diferents algorismes de planificació.

- *BFS (Breadth-First Search):* és l'algorisme de planificació més senzill. Donat un node inicial o llavor, s'explora aquest node i se n'obtenen tots els seus veïns. Posteriorment, s'exploren els veïns, obtenint-ne també els seus veïns. L'algorisme continua amb els nous veïns obtinguts. Aquest algorisme també l'anomenarem FIFO (*First In First Out*) ja que actua com una cua on el primer node en entrar és el primer en sortir. Inicialment, el node llavor es posa a la cua. Com que és el primer element afegit a la cua, serà el primer en ser explorat. Una vegada explorat el node inicial, es descobreixen els seus veïns, que són afegits a la cua. El següent node a explorar serà el primer dels veïns trobat. Els veïns d'aquest node seran també afegits a la cua i seran explorats una vegada s'hagin explorat tots els veïns del node inicial.

- *DFS (Depth-First Search):* l'algorisme comença amb una llavor i explora cada branca fins al final abans de tornar enrere. Aquest algorisme també l'anomenarem FILO (*First In Last Out*) ja que es comporta com una pila, on el primer node en entrar és l'últim en sortir. Inicialment, el node llavor se situa a la cua. Com que no hi ha cap altre element, el node inicial és el primer en ser explorat. Una vegada explorat el node inicial, els seus veïns són afegits a la pila. El següent node a explorar és l'últim que s'ha afegit, és a dir, l'últim dels veïns trobats. Els veïns d'aquest node són afegits a la pila i seran explorats, per tant, abans que els veïns del node inicial.

- *Selecció aleatòria(rand):* l'algorisme consisteix en seleccionar un node de manera ale-

atòria de la llista de nodes pendents a explorar. Aquest algorisme permet descobrir els nodes del graf d'una manera menys estricte que BFS o DFS sense la necessitat d'incrementar el temps de càlcul necessari per realitzar la selecció com els següents algorismes.

- *Greedy*: aquest algorisme consisteix en seleccionar, de la llista de nodes pendents per explorar, el que tingui el grau més gran. Com que el grau real d'un node no es coneix fins que s'ha explorat tot el graf, el valor que es fa servir és el grau del node dins del subgraf ja explorat. Al seleccionar primer els nodes amb grau més gran, aquest algorisme intenta millorar la velocitat de descobriment dels nodes del graf. La implementació d'aquest algorisme afegeix un temps de càlcul a la selecció dels nodes a explorar ja que, per cada decisió, cal calcular el grau de tots els nodes de la llista (o bé aprofitar els càlculs de l'elecció anterior i modificar-los amb la informació del nou node explorat).

- *Lottery*: aquest algorisme uneix el component d'aleatorietat que ofereix la selecció aleatòria amb la idea de seleccionar el node de grau més gran que introdueix l'algorisme *greedy*. L'algorisme *lottery* selecciona de la llista de nodes pendents el següent node a explorar de manera proporcional al seu grau. Per tant, un node amb un grau elevat és més probable que sigui elegit com a següent que un node amb un grau petit. Tot i així, l'algorisme no descarta l'elecció de nodes amb grau petit, compensant en part el biaix del grau mitjà del graf obtingut que genera el *greedy*.

Tots aquests algorismes comparteixen tres característiques importants: no necessiten informació prèvia sobre el graf per poder ser aplicats, no descarten nodes ja explorats i parteixen d'una llavor inicial. La primera de les característiques els fa útils per al *crawling* d'OSN de les quals no en tenim coneixement a priori. La segona suposa que tota la informació recollida és utilitzada i garanteix que es puguin realitzar comparacions justes entre els diferents algorismes. La tercera de les característiques fa que aquests mètodes siguin coneguts com a mètodes de bola de neu (o *snowball*). El terme s'utilitza, en general, per designar les tècniques de recollida de mostres on es parteix d'un individu o conjunt d'individus i es va ampliant la població a partir dels individus relacionats amb els anteriors. Aquest enfoc, a part de la introducció dels esmentats biaixos en les dades recollides, també presenta el problema que només és capaç de descobrir nodes que es trobin en el mateix component connex que el node inicial. Per tant, és possible que hi hagi nodes del graf que no puguin arribar a ser mai descoberts.

Hi ha un altre algorisme que cal esmentar tot i que no ha estat implementat en el *crawler* per no complir la condició de no necessitar informació prèvia sobre el graf. Anomenarem *greedy* hipotètic a l'algorisme *greedy* aplicat sobre els valors reals dels graus dels nodes. Aquest algorisme no pot ser implementat ja que no es coneix quin és el grau d'un node abans de explorar-lo però és útil a nivell teòric per definir un punt de referència.

5.1.4 El dispositiu d'emmagatzemament

La informació extreta pel *parser* és emmagatzemada en una base de dades relacional per al seu posterior anàlisi. Aquesta informació consta, principalment, de dos tipus de dades: dades sobre els usuaris (atributs dels nodes del graf) i dades sobre les relacions entre usuaris (les arestes i els seus atributs). Aquests dos tipus de dades es troben emmagatzemades en les dues taules principals de la base de dades (_Users i _Rel).

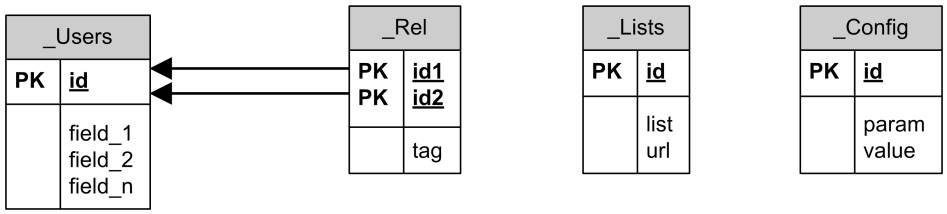

Figura 5.2: Diagrama relacional.

El nombre de camps de cada una de les taules variarà depenent de l'OSN analitzada. Totes les OSN comparteixen una estructura comú i tenen uns camps variables que depenen del tipus d'informació que se'n pugui extreure de cada una.

A més, la base de dades disposa d'altres taules auxiliars (_Lists i _Config) que permeten la recuperació de sessions de *crawling* de manera automàtica.

5.1.5 Altres característiques del web-crawler

A part de les característiques bàsiques descrites anteriorment, el *crawler* desenvolupat disposa d'altres funcionalitats.

- *Creació de logs:* el *crawler* permet activar la creació de *logs* amb diferents nivells de verbositat. A nivell 0, la funció es troba desactivada i no es genera cap tipus de *log*. A nivell 1, es crea un *log* amb informació de la configuració del *crawler*, el moment en el que s'explora cada node, els enllaços que es van afegint a la cua del planificador i certa informació de control com ara la mida de la llista de nodes pendents a explorar i els errors que es produeixen. El nivell 2 ofereix, a part de tota la informació del nivell 1, informació sobre el número de nodes explorats i descoberts, el número de relacions descobertes i el grau mitjà del graf en cada moment. Fer servir el nivell 2 suposa un increment considerable del temps d'execució del *crawler* ja que, per cada node explorat, s'han de calcular les mètriques abans esmentades. Per aquest motiu, aquest nivell només serà usat quan aquesta informació sigui necessària (per exemple, per a l'anàlisi dels diferents algorismes de *crawling*). El nivell 3 representa el nivell més alt i està reservat per a tasques de *debugging* o d'ajust de paràmetres. Si es fa servir amb el *parser* per *blogs* amb heurística, mostra totes les llistes d'enllaços trobades a cada pàgina amb les seves corresponents puntuacions segons l'heurística definida.

- *Sessions:* donat que una sessió de *crawling* pot ser molt llarga, el *crawler* disposa de la funció de guardar una sessió i recuperar-la posteriorment. Quan es recupera una sessió guardada, el *crawling* continua des del punt on es va deixar quan va ser guardada.

- *Exportació:* els grafs resultants del *crawling* són emmagatzemats a la base de dades, des d'on poden ser exportats en els formats *dot* i *GML* per al seu tractament. Aquesta funcionalitat és de vital importància per poder analitzar les dades obtingudes amb programes específics de tractament de grafs i per a la seva visualització.

- *Recollida d'informació sobre l'usuari:* el *crawler*, en el seu mode de funcionament bàsic, només recull la informació bàsica dels usuaris i les seves relacions (en concret, només recull la informació que es pot obtenir de la pàgina on es visualitzen els amics d'un usuari). En certs casos, com ara la definició de les llavors de l'algorisme d'agregació o bé la comprovació de les correspondències trobades, pot ser necessari obtenir més informació de cada un dels usuaris. Per aquest motiu, el *crawler* també disposa d'un mode de recollida d'informació més exhaustiu, que recol·lecta més informació sobre cada un dels usuaris tot analitzant la pàgina del seu perfil.

5.2 Representació

Una vegada s'han obtingut els grafs socials, és útil poder-los analitzar visualment. Per fer-ho, s'han fet servir diferents eines de visualització i tractament de grafs depenent del propòsit de la visualització i de la mida del graf a tractar.

- *GraphViz[8]:* és una eina de visualització de grafs que disposa de diferents programes per a realitzar-ne els *layouts*. *GraphViz* llegeix un fitxer d'entrada amb la descripció del graf en text pla i permet generar com a sortida un altre fitxer de text amb el *layout* del graf o bé generar directament una imatge en algun dels múltiples formats que suporta. Aquesta eina ha estat utilitzada per realitzar tant *layouts* de grafs de mida mitjana, que seran posteriorment visualitzats amb alguna de les altres eines que s'expliquen a continuació, com per generar directament imatges de grafs petits[3].

  Dels diferents programes que permeten generar els *layouts*, s'ha utilitzat, principalment, *Neato*. *Neato*[37] genera els *layouts* fent servir l'algorisme Kamada-Kawai[27], un algorisme de dibuix de grafs dirigit per forces. Els algorismes dirigits per forces es basen en interpretar cada node com a un cos físic real (amb massa i càrrega elèctrica) i cada aresta com una molla. D'aquesta manera, es defineixen les forces que actuen sobre el graf i se simulen com si fos un sistema físic, intentant buscar un estat de mínima energia global. Aquest estat de mínima energia coincideix amb els criteris d'estètica visual del graf, aconseguint minimitzar la superposició dels nodes i els encreuaments de

[3]Considerem un graf com a petit si té de l'ordre de 100 nodes o menys

les arestes i millorant la simetria del graf. Podem veure un exemple d'una imatge d'un graf generada amb *Neato* a la Figura 5.3.

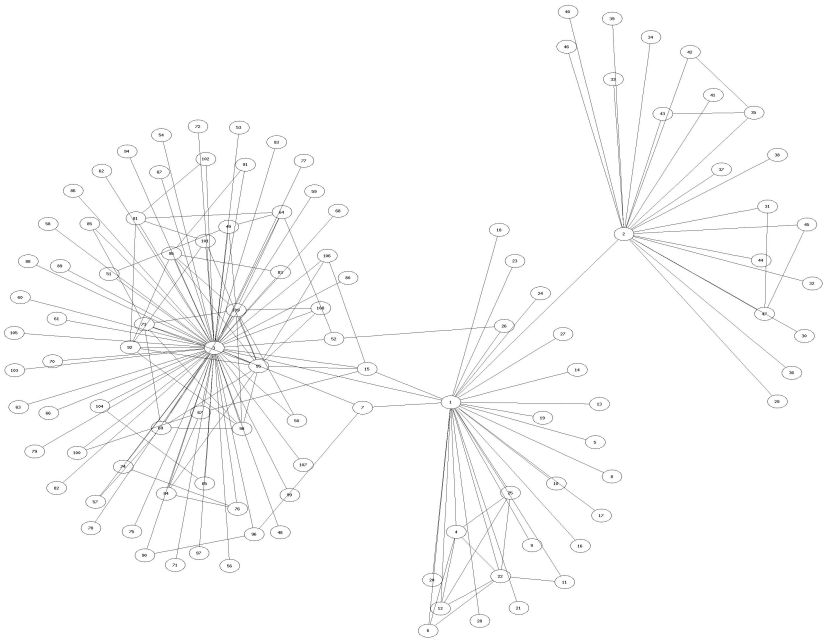

Figura 5.3: Exploració de la xarxa *Lastfm* amb l'algorisme FIFO. Imatge generada amb *Neato*.

- *Tulip[2]:* és un programa de visualització de grafs que permet treballar amb grafs grans (de l'ordre d'un milió de nodes). És capaç de llegir diferents formats de grafs (entre els quals hi ha *GML* i *dot*) i permet visualitzar-los de manera interactiva, oferint la possibilitat de navegar pels grafs. *Tulip* també permet crear o modificar el *layout* d'un graf aplicant diferents algorismes o bé permetent que l'usuari modifiqui les posicions dels nodes manualment. L'aplicació de filtres de colors segons alguna mètrica del graf permet facilitar la interpretació dels grafs, a part de millorar notablement l'efecte visual dels mateixos. Els resultats obtinguts es poden exportar en forma d'imatges o bé en diferents formats de descripció de grafs (GML o tlp, el format propi de *Tulip*).

- *VisOne[12]:* és un programa d'anàlisi i visualització de xarxes socials. Presenta unes característiques similars a *Tulip*: permet importar diferents formats de grafs (entre els quals hi ha *GML*), visualitzar-los de manera interactiva, crear o modificar el *layout* del graf aplicant diferents algorismes o bé permetent modificacions manuals de l'usuari, aplicar diferents mètriques sobre els nodes del graf i exportar els resultats forma d'imatges o bé en un format de descripció de grafs (*graphML*). La raó per la qual es fa servir tant *Tulip* com *VisOne* és que tots dos ofereixen alguns algorismes de generació de *layout* i de càlcul de mètriques que l'altre no ofereix. Podem veure diversos exemples de grafs generats amb *VisOne* al Capítol 6.

5.3 Anàlisi de la informació

Per tal d'analitzar les dades obtingudes s'ha fet servir, principalment, Matlab. Addicionalment, s'ha utilitzat algun dels programes comentats a la Secció 5.2 i altres programes especialitzats en anàlisi de xarxes.

- *VisAnt[49]:* és una plataforma interactiva per a la visualització i anàlisi de grafs. Encara que està especialitzada en anàlisi de xarxes biològiques, disposa d'un conjunt de funcions genèriques que són aplicables a qualsevol tipus de graf. Tot i que aquesta eina permet la visualització de grafs, durant el desenvolupament d'aquest projecte s'ha utilitzat principalment en l'anàlisi de dades, deixant la visualització a càrrec de les aplicacions comentades a la Secció 5.2. En concret, *VisAnt* s'ha utilitzat per comprovar si la distribució dels graus dels nodes segueix una llei de la potència. També ha estat utilitzat per calcular tot tipus de mètriques de grafs grans fent servir al seu mode de funcionament *batch*, que permet programar un conjunt de tasques a realitzar que s'aniran executant seqüencialment.

- *Matlab[32]:* ha estat l'eina més utilitzada per realitzar l'anàlisi de les dades obtingudes. Matlab és un llenguatge d'alt nivell que està especialitzat en anàlisi i visualització de dades, desenvolupament d'algorismes i realització de càlculs intensius, motiu pel qual és el llenguatge ideal per a realitzar l'anàlisi de les dades obtingudes de les diferents OSNs.

  Per tal de realitzar aquest anàlisi, s'han elaborat tres conjunts d'*scripts*. El primer conjunt, disposa de les funcions necessàries per analitzar els resultats dels *crawlings* i les alteracions introduïdes pels diferents algorismes de planificació. El segon conjunt, està format per la implementació de l'algorisme d'agregació i les funcions que permeten analitzar-ne el seu funcionament. El tercer conjunt està compost per funcions auxiliars desenvolupades per realitzar tasques diverses al llarg d'aquest projecte.

  El conjunt d'*scripts* per a l'anàlisi del *crawling* permet generar les gràfiques de l'evolució del grau mitjà, el número de nodes descoberts i el número de relacions descobertes segons el número de nodes explorats en cada moment. Aquestes gràfiques permeten analitzar com ha estat el procés d'obtenció de cada un dels grafs de les OSN analitzades. Els *scripts* reben com a entrada els *logs* de l'exploració (realitzats amb nivell 2), n'extreuen la informació necessària i la presenten de manera gràfica.

  El segon conjunt d'*scripts* està centrat en l'algorisme d'agregació de grafs. Conté l'algorisme en sí mateix (que es pot veure en detall a l'Annex A), funcions per mostrar resultats estadístics sobre el resultat de l'agregació, funcions d'importació i exportació de grafs i fitxers de *test*. Els fitxers de *test* permeten realitzar proves de l'algorisme amb grafs generats artificialment, de tal manera que es pot saber la correspondència correcta per a tots els nodes dels grafs agregats i per tant, quantificar amb precisió el grau d'encert de l'algorisme d'agregació. L'ús d'aquests grafs artificials també ha estat

útil a l'hora de determinar els valors dels paràmetres de l'algorisme i d'estudiar-ne la seva eficiència en diferents circumstàncies.

El tercer conjunt conté tot de funcions auxiliars que s'han anat necessitant durant el desenvolupament del projecte. Entre d'altres, hi trobem funcions de lectura de *logs*, creació de gràfiques per a mostrar els resultats i creació de grafs artificials.

CAPÍTOL 6

Anàlisi de les dades obtingudes

En aquest capítol s'exposen els resultats d'aplicar les tècniques explicades al Capítol 4 sobre els grafs obtinguts explorant les diferents OSN comentades al Capítol 3 amb el *crawler* desenvolupat. En primer lloc, es presenten els diferents grafs obtinguts i se n'exposen les característiques més destacades. Seguidament, es realitza una anàlisi dels diferents grafs en base a les mesures explicades a la Secció 4.1. Per últim, s'analitza el resultat d'agregar alguns dels grafs obtinguts.

6.1 Les dades recollides

El conjunt de dades recollides que s'analitzarà consta de 26 grafs extrets de les xarxes socials *online* descrites a la Secció 3.1. Per a cada una de les xarxes, s'han realitzat 5 exploracions corresponents als 5 algorismes de planificació explicats a la Secció 5.1.3. Totes les exploracions, exceptuant la de la xarxa *Typepad*, tenen com a llavor inicial l'usuari *Chess* (o *Chechar*), un usuari amb alta representació a les xarxes socials més populars que es va oferir a fer de conillet d'Índies per a les proves. En totes les exploracions realitzades, el *crawler* va ser configurat per esperar un temps aleatori entre 1 i 10 segons entre cada una de les peticions realitzades. Algunes de les exploracions van ser realitzades fent servir la xarxa Tor.

L'ús de la xarxa Tor per realitzar els *crawlings* permet ocultar l'origen de les consultes a les OSN però també comporta un augment del temps necessari per realitzar l'exploració. L'increment de temps que es produeix és poc notable quan estem fent servir temps d'espera elevats entre peticions o bé en iteracions avançades dels algorismes *greedy* o *lottery*, que ja

afegeixen un temps de càlcul a l'elecció del següent node . En canvi, aquest temps es torna força crític si el que volem és aconseguir descobrir el major nombre de nodes possible en el menor temps.

OSN	Algorisme planificació	Número nodes explorats	Número nodes descoberts	Número relacions descobertes	Grau mitjà
Flickr	Fifo	102	25255	34085	336.6
	Filo	162	29590	34390	213.4
	Greedy	102	46146	83950	841.7
	Lottery	105	37128	49994	477.7
	Rand	101	40331	52030	516.4
Lastfm	Fifo	109	8039	8622	80.6
	Filo	96	10965	11689	122.9
	Greedy	115	44194	53069	472.3
	Lottery	95	14106	21877	243.8
	Rand	106	11876	12804	121.8
Twitter	Fifo	106	223329	245307	2317.7
	Filo	104	156131	193761	1867.6
	Greedy	94	82723	167210	1810.3
	Lottery	121	358945	801162	6624.4
	Rand	100	154073	251181	2513.3
Blogs sense heurística	Tots	28	35	58	3.8
Blogs amb heurística	Fifo	111	477	506	5.7
	Filo	102	276	460	6.5
	Greedy	105	217	344	5.0
	Lottery	110	458	533	6.1
	Rand	110	382	438	5.2
Typepad	Fifo	124	2438	6671	64.3
	Filo	122	2932	4608	39.14
	Greedy	109	5545	11918	138.7
	Lottery	108	1785	5115	54.2
	Rand	113	3307	6553	61.1

Taula 6.1: Resum dels grafs obtinguts.

A la Taula 6.1 es mostren les principals mesures dels grafs obtinguts. Per cada graf, hi trobem la xarxa social analitzada, l'algorisme de planificació utilitzat, el número de nodes explorats i descoberts, el número de relacions descobertes i el grau mitjà dels nodes. Anomenarem nodes explorats els nodes que han estat visitats pel *crawler* i dels quals en coneixem, per tant, tots els seus contactes. En canvi, anomenarem nodes descoberts aquells nodes dels quals en coneixem la seva existència (són contactes de nodes explorats) però que encara no han estat visitats pel *crawler*. El número de relacions descobertes fa referència al número d'arestes del graf.

El primer que podem apreciar observant aquests resultats és la gran diferència entre el número de nodes explorats i el número de nodes descoberts, el que és una mostra del fenomen del *small world*. En el cas més extrem (*Twitter* amb *lottery*), 121 nodes explorats ens permeten descobrir-ne 358945.

Un altre fet destacable és l'elevat grau mitjà que presenten els nodes explorats. Tot i que tots els algorismes de *crawling* utilitzats produeixen sobreestimacions del grau mitjà i que no es poden extrapolar aquests resultats al graf complet ja que el graf explorat és molt petit, el grau mitjà dels nodes explorats és sorprenentment alt. A tall de curiositat, el node explorat amb grau més elevat de la xarxa *Twitter* té de l'ordre de 50000 veïns (30000 seguidors i 20000 persones a les quals segueix).

Així mateix, podem veure com fer servir *greedy* com a algorisme de planificació no sempre comporta descobrir més quantitat de nodes amb el mateix nombre de nodes explorats. El motiu el trobem en la localitat de les decisions preses per l'algorisme, que tria el node que més veïns ja explorats té i no pas el que té més veïns no descoberts (aquesta informació la desconeix). A més, encara que fos possible triar el node que més veïns no descoberts tingués, aquesta podria no ser una bona decisió després d'unes quantes iteracions del *crawler*.

Un altre dels fets que podem observar és que els grafs explorats són molt petits respecte al graf complet de cada xarxa. Indicis que ens ho demostren són la diferència entre el número de nodes explorats i els descoberts o bé la diferència del grau mitjà dels nodes explorats amb el quocient de les relacions descobertes entre nodes descoberts (que hauria de ser 2 si s'hagués explorat tot el graf).

El *parser* de *blogs* sense heurística presenta uns resultats força peculiars. L'ús d'aquest *parser* fent servir com a node inicial l'usuari 0 produeix un sol component connex de 28 nodes. L'ús de diferents algorismes de planificació fa diferir l'ordre amb el qual s'exploren els nodes però obté els mateixos resultats. La diferència entre el número de nodes descoberts i el número de nodes explorats en aquest cas és deguda a enllaços que no poden ser processats (per exemple, urls incorrectes o que demanen autenticació). En aquests casos, els nodes comptabilitzen com a descoberts (ja que s'han trobat en l'exploració) però no com a explorats, ja que no s'han pogut analitzar correctament.

6.2 Anàlisi de les dades recollides

Al Capítol 4 s'exposaven algunes de les mesures que es poden aplicar als grafs per tal d'extreure'n informació. En aquesta secció, es mostren els resultats de calcular aquestes mesures sobre els grafs obtinguts amb el *crawler* i s'analitzen les seves implicacions. Per tal de realitzar els càlculs, s'han fet servir els subgrafs induïts pel conjunt de vèrtex explorats i que anomenarem, a partir d'ara, subgrafs explorats. Per tant, per cada un dels grafs analitzats, les mesures es troben calculades sobre el graf format pel conjunt de nodes explorats i les relacions entre aquests nodes. D'aquesta manera s'aconsegueix tant que la realització de tots els càlculs es pugui realitzar en un temps raonable com que els grafs puguin ser analitzats de manera visual.

És important tenir en compte que moltes de les diferències entre els diferents algorismes de planificació que veurem es veuen accentuades pel fet que els subgrafs explorats són petits respecte al graf complet de la xarxa analitzada. L'exploració de més nodes del graf reduiria aquestes diferències fins al punt que, amb tota la xarxa explorada, aquestes serien nul·les (els grafs obtinguts serien iguals, com en el cas dels *blogs* sense heurística).

6.2.1 Diàmetre

Els resultats de calcular el diàmetre dels grafs obtinguts de les diferents OSN amb els diferents algorismes de planificació es poden observar a la Taula 6.2. Observant aquests resultats podem comprovar com els diàmetres obtinguts depenen fortament de l'algorisme de planificació utilitzat. En la majoria de xarxes analitzades (totes excepte els *blogs*), el diàmetre del graf obtingut permet ordenar els planificadors de la mateixa manera: $FIFO \leq greedy < lottery < rand \ll FILO$. En el cas dels *blogs*, l'algorisme *rand* obté un diàmetre un punt menor que el del *lottery*.

Els grafs obtinguts fent servir l'algorisme *FIFO* sempre presenten un diàmetre menor que els obtinguts amb la resta d'algorismes. Això és degut a que aquest algorisme fa que s'explorin primer els nodes més propers al node inicial i, per tant, per a un mateix número de nodes explorats el diàmetre tendirà a ser mínim. L'efecte contrari el trobem amb els grafs obtinguts amb l'algorisme de planificació *FILO*, que presenten els diàmetres més grans de manera diferenciada (en tots els casos exceptuant els *blogs* supera el doble del valor obtingut per l'algorisme aleatori). El cas més evident d'aquest fenomen el trobem analitzant la xarxa *Twitter*. Fent servir *FILO*, el graf obtingut presenta un diàmetre de 2, el que indica que entre cada parell de nodes podem trobar un camí que els uneixi passant només per un tercer node. Tot i que el resultat pot sorprendre inicialment, analitzant el graf podem veure que l'usuari fet servir com a llavor té 118 contactes en aquesta xarxa. Com que el graf obtingut està compost dels 109 nodes explorats que s'han escollit amb *FIFO*, veiem que els nodes explorats són tots veïns del node inicial. Per aquest motiu, donats dos nodes qualssevol, sempre hi haurà un camí entre ells de distància 2 que passarà per aquest node inicial.

OSN	Algorisme planificació	Número nodes explorats	Diàmetre graf explorat	Coeficient agrupament mitjà	Grau mitjà
Flickr	Fifo	102	3	0.343	4.882
	Filo	162	32	0.026	2.198
	Greedy	102	3	0.636	37.294
	Lottery	105	8	0.032	3.086
	Rand	101	10	0.025	2.535
Lastfm	Fifo	109	4	0.167	3.101
	Filo	96	25	0.065	2.28
	Greedy	115	9	0.457	21.67
	Lottery	95	10	0.171	4.2
	Rand	106	12	0.007	2.132
Twitter	Fifo	106	2	0.497	7.047
	Filo	104	24	0.352	5.795
	Greedy	94	3	0.776	62.894
	Lottery	121	6	0.188	6.496
	Rand	100	9	0.077	3.089
Blogs sense heurística	Tots	28	3	0.047	3.143
Blogs amb heurística	Fifo	111	6	0.014	2.288
	Filo	102	17	0.182	3.922
	Greedy	105	12	0.138	3.429
	Lottery	110	13	0.068	2.436
	Rand	110	12	0.033	2.382
Typepad	Fifo	124	3	0.605	21.048
	Filo	122	13	0.062	2.754
	Greedy	109	2	0.748	58.862
	Lottery	108	4	0.477	13.611
	Rand	113	4	0.327	6.159

Taula 6.2: Diàmetre i coeficient d'agrupament dels subgrafs explorats.

El diàmetre obtingut fent servir l'algorisme *greedy* és una mica superior que l'obtingut amb *FIFO*. Això s'explica pel fet que aquest algorisme tria com a següent node a explorar aquell que té un grau més elevat, és a dir, que té un nombre més alt de veïns ja explorats. Per

l'estructura dels grafs obtinguts, aquest node és poc probable que es trobi en un dels extrems de la geodèsica més llarga del graf i, per tant, explorar-lo normalment no comportarà augmentar el diàmetre del graf. A la Figura 6.1 es pot veure un exemple d'aquest fenomen. El node 1 s'ha fet servir com a node inicial del *crawling*. A l'explorar-lo, s'han descobert els nodes 2, 3, 4, 5 i 6. En aquest moment, tots els nodes no explorats tenen grau 1 i per tant, l'algorisme de planificació en triarà un seguint algun altre criteri (en el nostre cas, seguirà una política FILO). S'explora per tant el node 6, descobrint tres nodes nous (7, 8 i 9) i una relació amb un node ja existent (el node 5). Ara, l'algorisme de planificació triarà el node 5 com a següent node a explorar ja que és el que té un grau més gran entre els pendents a explorar. Abans d'explorar el node 5, el diàmetre del graf és 3 (les geodèsiques més llargues són les que uneixen els nodes 2, 3 i 4 amb els 7, 8 i 9). Podem veure com, explorant el node 5, el diàmetre del graf en cap cas podrà augmentar ja que, com a molt, la distància entre els nous nodes descoberts i els anteriors serà de 3, que ja era el diàmetre abans d'explorar el node 5.

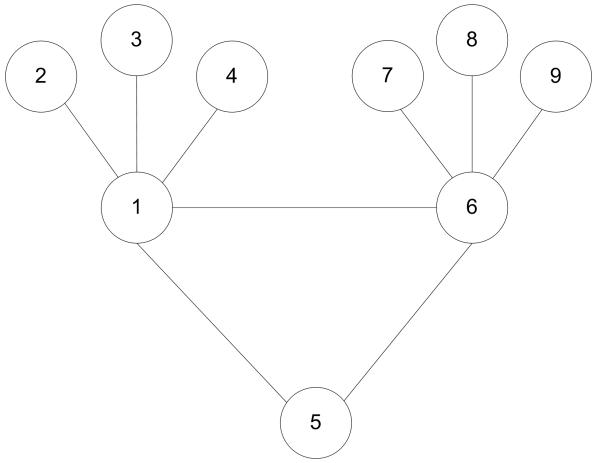

Figura 6.1: Justificació del diàmetre dels grafs obtinguts amb *greedy*.

Amb l'algorisme *lottery*, el següent node a explorar es tria aleatòriament amb una probabilitat proporcional al seu grau i, per tant, també premia els nodes amb més grau però d'una manera menys estricte que el *greedy*. El fenomen explicat anteriorment succeeix en menor grau al fer servir l'algorisme *lottery* i, per tant, els grafs obtinguts amb aquest mètode presenten un diàmetre lleugerament superior als obtinguts amb *greedy*.

L'algorisme *rand* tria el següent node a explorar de manera aleatòria fent servir una distribució uniforme. Per tant, tots els nodes tenen la mateixa probabilitat de ser escollits a cada iteració i el fenomen explicat succeeix encara en menor grau que en el *lottery*. Per aquest motiu, els diàmetres obtinguts amb aquest algorisme són superiors als obtinguts amb *lottery*.

Com s'ha comentat anteriorment, aquests resultats han estat calculats a partir del subgraf induït pels nodes explorats. Els nodes que han estat descoberts però no explorats no han estat tinguts en compte per a aquests càlculs. Tot i així, podem afirmar que tots els nodes

descoberts són veïns de nodes que han estat explorats (ja que d'altra manera, no haurien pogut ser descoberts). Per tant, els diàmetres dels grafs complets que s'han obtingut poden diferir, com a molt, en dos salts respecte als resultats mostrats.

6.2.2 Agrupament

Com hem vist a la Secció 4.1.2, els grafs socials es caracteritzen per presentar un coeficient d'agrupament elevat. A la Taula 6.2 hi podem trobar els coeficients d'agrupament mitjans dels grafs obtinguts i es pot observar com aquests són grans per als grafs obtinguts amb l'algorisme de planificació *greedy* però, en canvi, són molt petits en alguns dels altres casos (el graf de *Lastfm* amb *rand* o el dels *blogs* amb heurística obtingut amb *lottery*).

El fet que els grafs obtinguts amb *greedy* presentin un coeficient d'agrupament més elevat que la resta pot resultar, inicialment, contraintuitiu, ja que com que els nodes d'aquests grafs tenen un grau més elevat, és d'esperar que el seu coeficient d'agrupament sigui més baix. El motiu pel qual això no succeeix és que el *greedy* tria com a proper node el que té més grau d'entre els ja explorats. El resultat d'aquesta elecció és, per tant, un node altament connectat als nodes ja explorats, el que acaba comportant un coeficient d'agrupament alt. Els grafs explorats amb *greedy* es caracteritzen per tenir tots els nodes amb coeficients d'agrupament força alts. Tenint en compte la distribució dels coeficients d'agrupament dels nodes segons el seu grau, aleshores sí que es compleix la nostra idea inicial: els nodes amb grau més baix presenten coeficients d'agrupament més elevats mentre que els nodes amb grau més alt presenten coeficients d'agrupament baixos. Un exemple d'aquest comportament el podem veure en el graf explorat amb *greedy* de la xarxa *Flickr* (Figura 6.2) on es poden apreciar dues comunitats amb nodes altament connectats dins de cada comunitat. La distribució dels coeficients d'agrupament segons el grau dels nodes del mateix graf es pot observar al gràfic mostrat en la Figura 6.3, on podem observar com els nodes amb grau més elevat tenen, generalment, coeficients d'agrupament més baixos que els nodes amb grau més baix.

L'exemple contrari, el d'un graf amb un coeficient d'agrupament molt baix, el trobem a la Figura 6.4. En aquest cas, només 6 nodes tenen un coeficient d'agrupament diferent de 0, és a dir, tenen veïns que es troben connectats entre ells. A més, podem veure com aquests 6 nodes formen 2 cliques màxims de mida 3. La resta dels nodes del graf presenten un coeficient d'agrupament de 0.

6.2.3 Mesures de centralitat de nodes

A la Secció 4.1.3 hem descrit algunes de les diferents mesures de centralitat. En aquest apartat, veurem els resultats de calcular aquestes mesures sobre els grafs obtinguts.

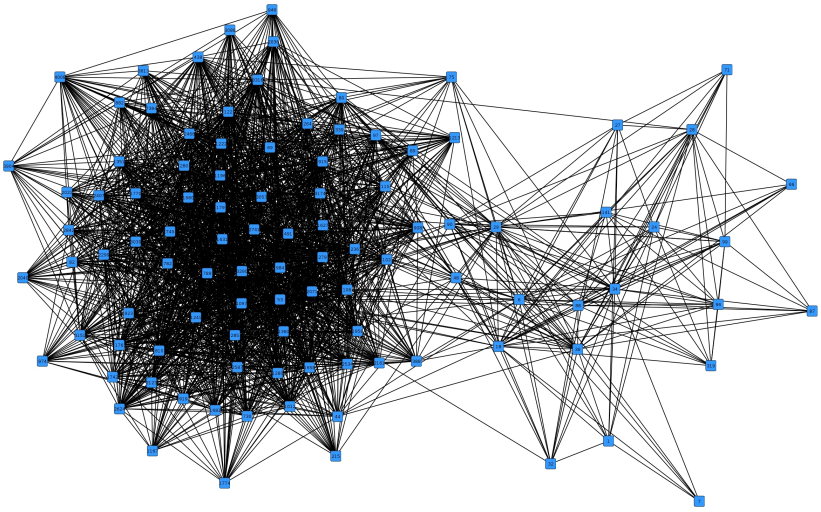

Figura 6.2: Graf explorat de la xarxa *Flickr* amb *greedy*.

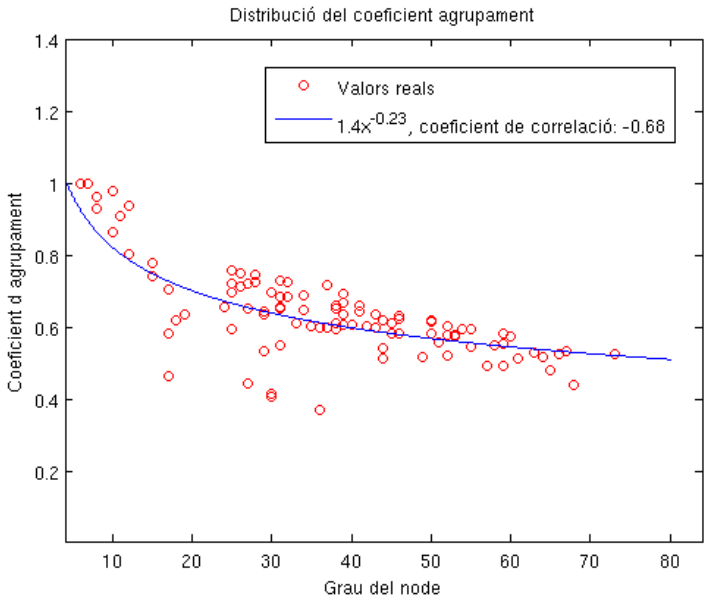

Figura 6.3: Distribució del coeficient d'agrupament (xarxa *Flickr* amb *greedy*).

Centralitat de grau

La centralitat de grau és la mesura de centralitat més simple ja que s'expressa, directament, com el grau del node. Com podem veure a la Taula 6.2, totes les xarxes analitzades (excepte els *blogs*) presenten un grau mitjà marcadament més elevat quan són explorades amb l'algorisme de planificació *greedy*. Aquest resultat s'obté ja que l'algorisme fa servir precisament el grau per a triar el proper node a explorar. Els grafs obtinguts amb *lottery* haurien de tenir, pel mateix motiu, un grau mitjà més elevat que la resta. Donat el component d'aleatorietat, aquest fenomen pot no ser observable a petita escala i, de fet, només s'observa en el graf

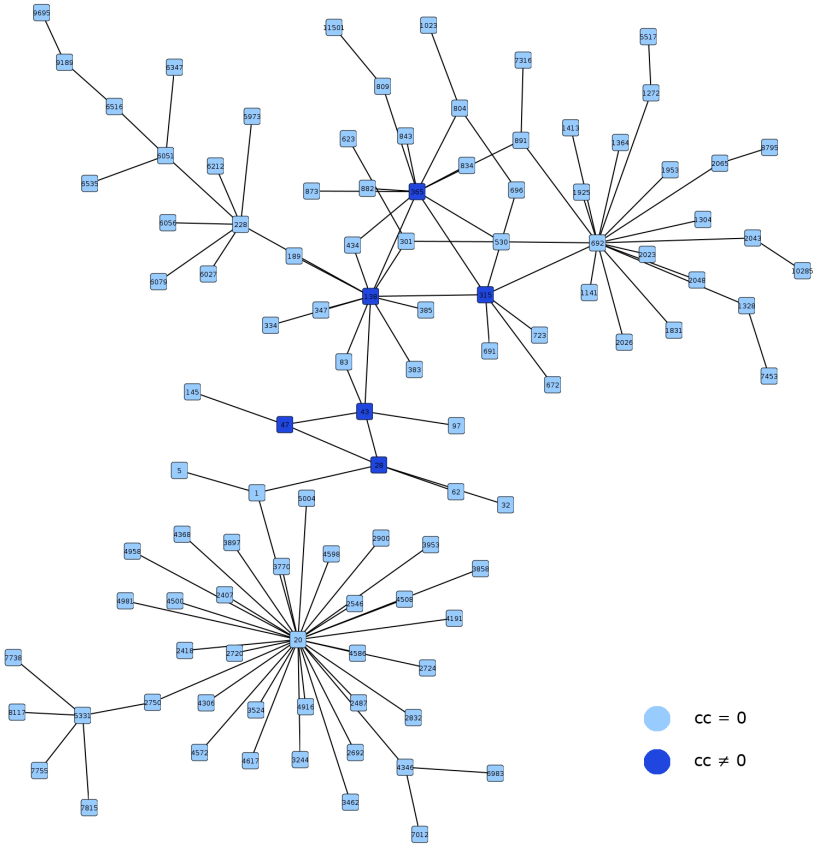

Figura 6.4: Graf obtingut de la xarxa *Lastfm* amb *rand*.

obtingut de la xarxa *Lastfm*. L'exploració d'un número més elevat de nodes hauria de fer que el grau mitjà obtingut amb l'algorisme de planificació *lottery* superés de manera més diferenciada l'obtingut pels algorismes *FIFO*, *FILO* i *rand*. Aquests 3 últims algorismes no tenen un efecte directe sobre el grau mitjà del graf obtingut ja que la selecció del següent node no es troba condicionada pel grau que aquest presenta.

Un altre dels fenòmens a analitzar relacionats amb el grau dels nodes és la seva distribució. Com hem vist a la Secció 2.2.2 la distribució dels graus dels nodes d'un graf social segueix una llei de la potència però els subgrafs d'un graf que segueix una llei de la potència poden no seguir-la. Els resultats en aquest sentit també depenen de l'algorisme de planificació utilitzat. Per una banda, els grafs obtinguts amb l'algorisme *greedy* no mostren aquesta distribució de graus i, de fet, ni tan sols compleixen que el número de nodes amb graus petits sigui més gran que el número de nodes amb graus grans. Una vegada més, aquest comportament és l'esperat per aquest algorisme ja que s'exploren els nodes amb grau més elevat. Podem veure un exemple de la distribució de graus de la xarxa *Twitter* amb l'algorisme de planificació *greedy* a la Figura 6.5. Per altra banda, les distribucions dels grafs obtinguts de la resta d'algorismes es poden ajustar de manera prou acurada a per la funció $f(x) = Ax^{-\lambda}$. Per exemple, es poden trobar valors per a A i λ per tal d'ajustar la distribució dels graus del graf obtingut de la xarxa *Twitter* amb *rand* obtenint un coeficient de correlació de -0.93 (veure

Figura 6.5). Les diferències obtingudes entre els algorismes *FIFO*, *FILO*, *lottery* i *rand* són molt petites i no varien entre les diferents xarxes analitzades.

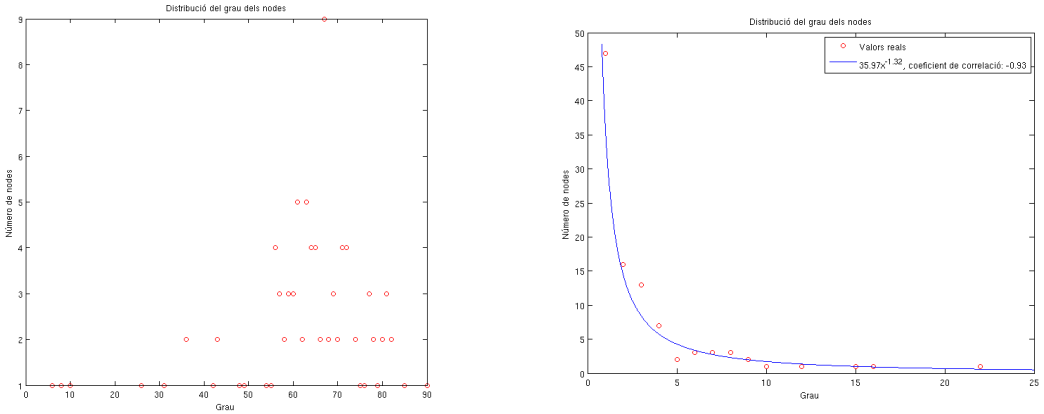

Figura 6.5: Distribució dels graus dels nodes de la xarxa *Twitter* (explorats amb *greedy*, a l'esquerra i *rand*, a la dreta).

Centralitat d'intermediació

La centralitat d'intermediació és una mesura de la freqüència en la que un node es troba entre un altre parell de nodes en la geodèsica que els interconnecta. Normalment, aquest valor serà elevat per al node inicial en les primeres iteracions del *crawler* funcionant amb *FIFO* i anirà disminuint conforme es vagin explorant nodes i descobrint relacions entre ells. Podem observar la centralitat d'intermediació de xarxa *Flickr* explorada amb *FIFO* a la Figura 6.6. En aquest cas, la centralitat del node inicial és del 37%, només superada pel node 2 (59%). La resta de nodes presenten una centralitat molt més baixa, de l'ordre de dues magnituds inferior. A més a més, hi ha 54 nodes que tenen una centralitat d'intermediació de 0.

Un altre dels algorismes que provoca unes centralitats d'intermediació particulars és el *FILO*. Les primeres iteracions del *crawler* amb *FILO* construeixen un graf explorat que consta d'un sol camí. En aquests moments, els nodes centrals del camí tenen una centralitat d'intermediació més elevada que va disminuint conforme ens anem apropant als extrems del camí. Al anar explorant més quantitat de nodes, el graf explorat comença a formar cicles i a anar disminuint les diferències entre la centralitat d'intermediació dels nodes. Podem veure un exemple de la mateixa xarxa explorada amb *FILO* a la Figura 6.7. Els nodes que es troben en un sol cicle simple tenen centralitats intermèdies més baixes i, en canvi, els nodes que permeten unir diferents cicles tenen les centralitats d'intermediació més elevades.

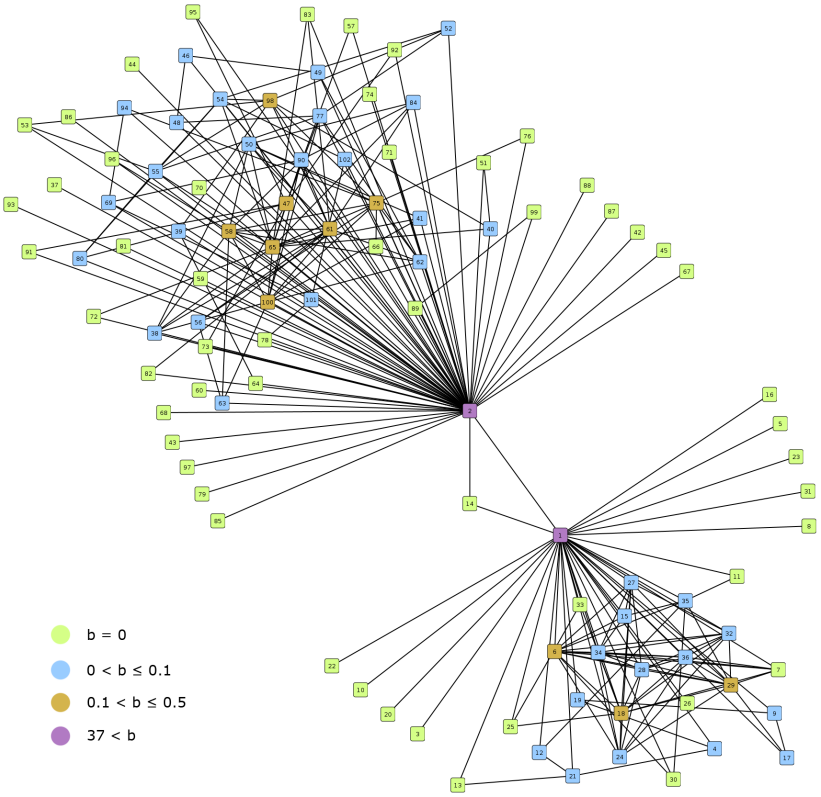

Figura 6.6: Centralitat intermèdia dels nodes de la xarxa Flickr explorats amb *FIFO*.

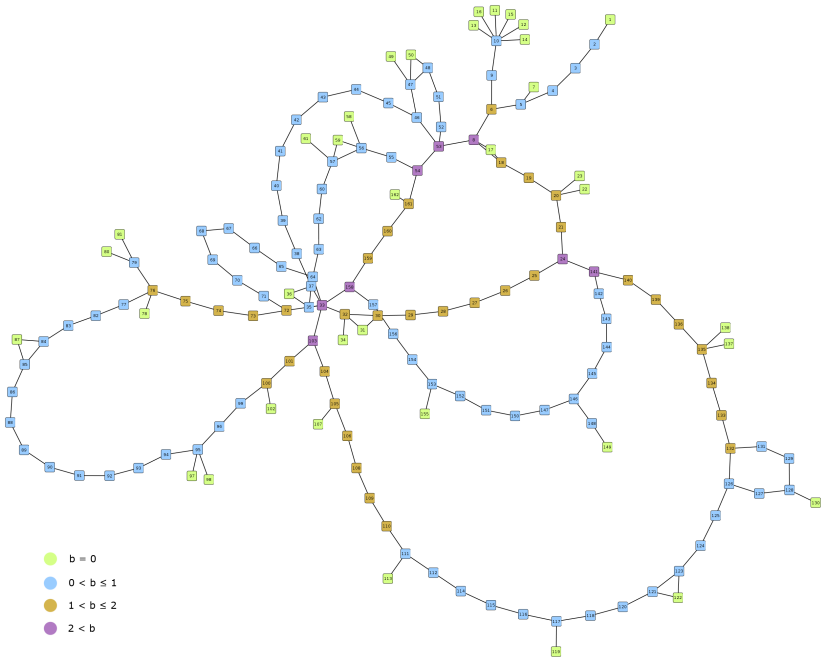

Figura 6.7: Centralitat intermèdia dels nodes de la xarxa *Flickr* explorats amb *FILO*.

Centralitat de proximitat

La centralitat de proximitat ens permet mesurar com de proper es troba un node de la resta de nodes del graf. Aquest valor també serà elevat per al node inicial quan fem servir un algorisme de planificació *FIFO* i tindrà un comportament similar amb els grafs obtinguts amb *FILO*, premiant els nodes que es troben a les interseccions entre els diferents cicles.

Podem observar les diferències entre la centralitat intermèdia i la de proximitat, per exemple, als nodes de grau 1. Aquests nodes sempre tenen una centralitat intermèdia molt baixa ja que òbviament, mai formaran part de la geodèsica que uneixi dos altres nodes però, en canvi, poden tenir una centralitat de proximitat elevada si es troben situats en la posició adequada. A la Figura 6.8 hi trobem el mateix graf que es comentava al parlar de centralitat intermèdia (Figura 6.6) però mostrant ara els valors de la centralitat de proximitat. Comparant-los, podem veure com els nodes de grau 1 que són veïns del node 2 (part superior del graf) tenen una centralitat de proximitat superior als nodes de grau 1 veïns del node 1 (part inferior del graf). Això és degut a que hi ha més quantitat de nodes en el clúster superior i, per tant, els nodes que es trobin en aquest clúster es troben més propers a més nodes del graf. En canvi, tant uns com els altres tenen una centralitat intermèdia igual a 0.

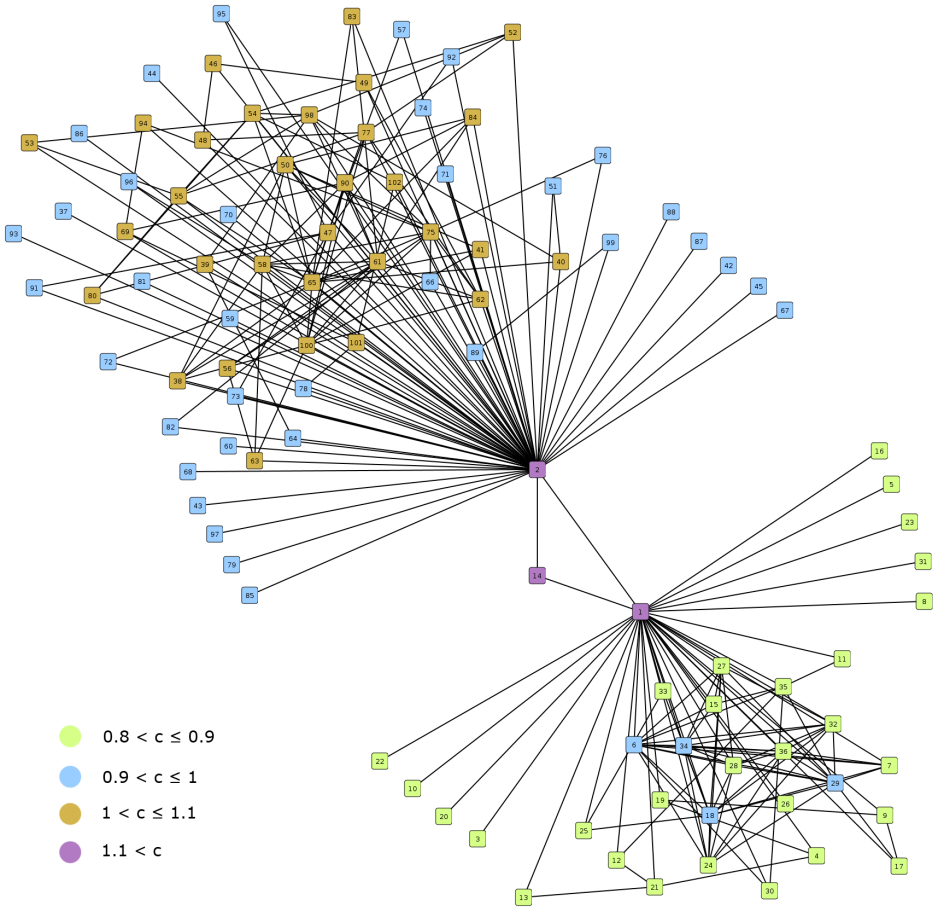

Figura 6.8: Centralitat de proximitat dels nodes de la xarxa *Flickr* explorats amb *FIFO*.

6.2.4 Detecció de comunitats

Com hem vist a la Secció 4.1.4, els individus d'una xarxa social es poden agrupar en comunitats. Les diferents comunitats es poden definir a diferents nivells de tal manera que, en el nivell més alt, tota la xarxa és una sola comunitat i, en el nivell més baix, cada un dels individus forma una comunitat per si sol. Entremig d'aquests dos extrems, disposem de múltiples nivells en els quals identificar les comunitats.

Els algorismes de detecció de comunitats aplicats als grafs objecte d'estudi obtenen resultats significatius amb els grafs obtinguts amb *FIFO*, *greedy*, *lottery* i *rand*, permetent obtenir comunitats amb diferents nivells de granularitat. En el cas dels grafs explorats amb *FILO* i donada la seva estructura (veure, per exemple, la Figura 6.7) les divisions en comunitats obtingudes no ens ofereixen cap informació significativa.

A les Figures 6.9 i 6.10 hi podem trobar dues divisions a diferents nivells del graf explorat amb *FIFO* de la xarxa *Lastfm*. La primera divisió (Figura 6.9) detecta tres comunitats al graf (marcades en blau, groc i verd a la figura), cada una de les quals té un nombre elevat de nodes. La segona divisió (Figura 6.10) consta de vuit comunitats diferents, una de les quals (marcada en verd), coincideix exactament amb la primera divisió. Les comunitats blava i groga de la primera divisió s'han particionat encara més i han format noves comunitats. En concret, podem veure com la comunitat groga ha derivat en dues comunitats i la blava en cinc.

6.3 Agregació

En aquest apartat s'analitzen els resultats de l'algorisme d'agregació d'informació que s'ha presentat a la Secció 4.2. Per tal d'avaluar com es comporta l'algorisme s'han realitzat quatre conjunts de proves diferents.

1. *Agregació de grafs aleatoris:* s'agreguen grafs aleatoris generats artificialment. Es parteix d'un graf aleatori inicial G del qual se'n fa una còpia G'. Es realitzen modificacions sobre el graf G' (per simular la introducció de soroll) i s'intenta agregar G amb G'.

2. *Agregació de grafs socials:* se segueix el mateix procediment però fent servir com a graf G algun dels grafs obtinguts de les OSN analitzades. Això permet avaluar el comportament de l'algorisme amb grafs socials que, com hem vist, tenen unes característiques diferents als grafs aleatoris.

3. *Simulació de l'agregació real:* fins ara, les proves realitzades es basaven en agregar dos grafs, G i G', que resultaven ser el mateix graf amb algunes modificacions. En aquest conjunt de proves, es recrea l'escenari d'una agregació real, definint dos grafs aleatoris G_1 i G_2 que comparteixen un conjunt de nodes però que en tenen molts d'altres de no comuns.

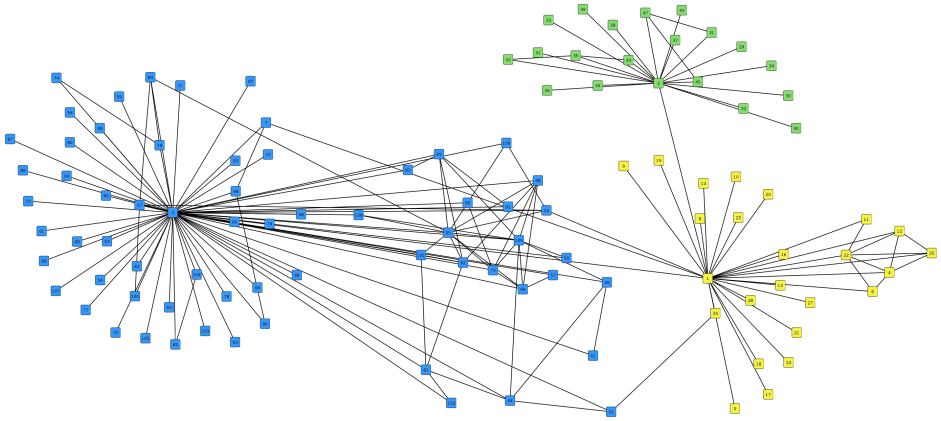

Figura 6.9: Detecció de comunitats (xarxa *lastfm* amb *FIFO*).

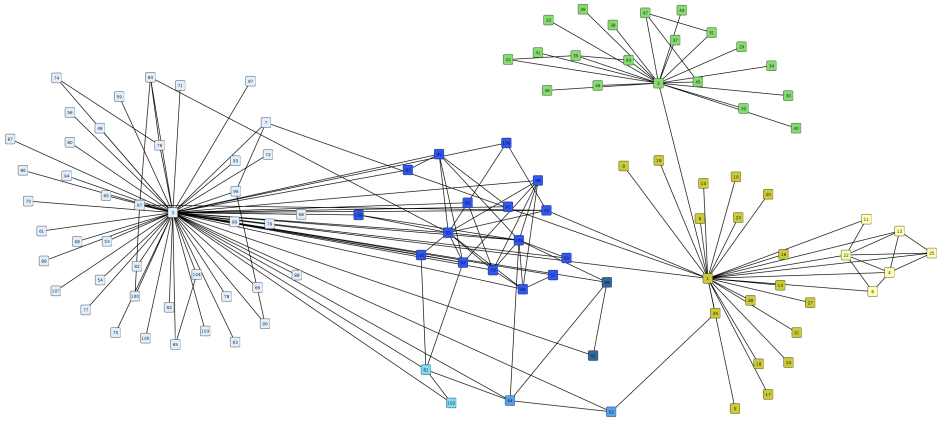

Figura 6.10: Detecció de comunitats (xarxa *lastfm* amb *FIFO*).

4. *Agregació real:* per últim, es realitza l'agregació dels grafs socials obtinguts de les diferents xarxes analitzades. En aquest cas, serà més difícil avaluar el resultat de l'agregació ja que no disposem d'informació completa sobre les correspondències reals entre els nodes de les diferents xarxes.

6.3.1 Agregació de grafs aleatoris

Un graf aleatori es pot construir fixant el número de nodes, n, i afegint les possibles arestes amb una probabilitat p. Aquest model es coneix amb el nom d'Erdos-Renyi. Una altra manera de construir grafs aleatoris és, donat el número de nodes, seleccionar parelles de nodes aleatoris entre les quals s'afegeix una aresta. Aquest segon mètode és equivalent al primer quan se seleccionen $\frac{p*n*(n-1)}{2}$ parelles de nodes.

Les proves d'agregació amb grafs aleatoris permeten analitzar el percentatge d'encert en la identificació de correspondències depenent de quin grau de similitud tenen els dos grafs agregats. Per tal d'efectuar aquestes proves, es genera un graf aleatori G d'ordre 100 (per similitud amb els grafs explorats de les OSN) i es crea un segon graf G' a partir de modificacions del graf G. Les modificacions efectuades poden ser de tres tipus: eliminació, addició o canvi d'a-

restes entre els nodes del graf (no es contempla la modificació del nombre de nodes). Fixat el número de modificacions a realitzar, es parteix del graf G i es van seleccionant, aleatòriament parelles de nodes entre les quals s'eliminen, s'afegeixen o es modifiquen les arestes existents. En el cas de l'eliminació d'arestes, només se seleccionen parelles de nodes que ja tenen una aresta que els uneix i en l'addició, aquelles que no en tenen. Les modificacions es poden fer entre qualsevol parella de nodes, afegint una aresta si no són veïns i eliminant-la si ja ho són.

El nombre de modificacions realitzades determinarà com de similars són els grafs agregats. Anomenarem percentatge de soroll introduït al nombre d'arestes modificades entre el nombre el nombre d'arestes comuns a G i G'. Per tant, el fet d'eliminar una aresta produirà més soroll que el fet d'afegir-ne. La modificació es comportarà de forma similar a l'addició en grafs poc densos i de forma similar a l'eliminació en grafs molt densos.

Una vegada generats G i G', se seleccionen aleatòriament 10 nodes que conformaran les correspondències inicials i s'executa l'algorisme[1]. El conjunt de correspondències final que retorna l'algorisme és analitzat posteriorment per determinar-ne la seva correcció.

Els resultats de repetir aquest experiment per a diferents nivells de soroll poden ser observats a la gràfica de la Figura 6.11. Cada un dels punts de la gràfica és, en realitat, la mitjana dels resultats obtinguts repetint l'experiment 25 vegades amb les mateixes condicions i diferents grafs aleatoris. Això permet minimitzar les possibles alteracions introduïdes per grafs o correspondències inicials especialment favorables o desfavorables al procés d'agregació.

Observant els resultats mostrats a la Figura 6.11 podem veure com l'algorisme és força resistent al soroll introduït per addició d'arestes. Amb 500 arestes afegides (quan el soroll és 1 s'han afegit tantes arestes com tenia originalment el graf) l'algorisme és capaç d'identificar correctament al voltant d'un 80% del nodes. Aquesta tolerància al soroll serà clau a l'hora d'agregar diferents grafs ja que, en aquests casos, les diferències entre els grafs a agregar poden ser molt elevades.

6.3.2 Agregació de grafs socials

El segon conjunt de proves s'ha realitzat repetint el procediment anterior però fent servir els grafs obtinguts de l'exploració de les diferents xarxes socials en comptes de grafs generats aleatòriament. Els resultats obtinguts depenen fortament del coeficient d'agrupament del graf fet servir. Amb grafs amb coeficients d'agrupament elevats (per exemple, els explorats amb *greedy*), els resultats de l'agregació són molt bons i demostren que l'algorisme és tolerant al soroll. En canvi, en grafs amb coeficients d'agrupament baixos, l'algorisme no és tant tolerant al soroll i els percentatges de reidentificació baixen considerablement.

[1] L'elecció del número de nodes inicials es realitza en base a un conjunt de proves prèvies que han determinat que aquest número és suficient per realitzar una agregació d'aquestes característiques. Els valors d'altres paràmetres de l'algorisme com el nombre màxim d'iteracions, el llindar d'excentricitat o el decrement d'aquest llindar també han estat determinats prèviament per tal d'optimitzar els resultats de l'agregació.

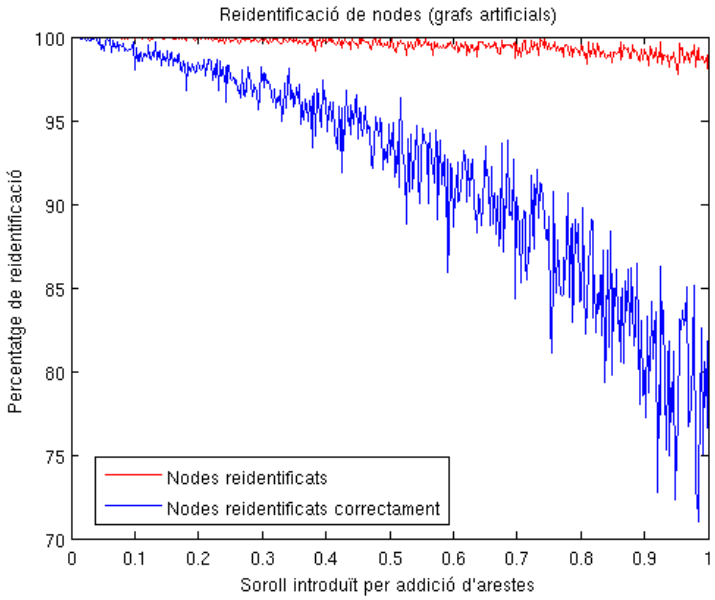

Figura 6.11: Agregació amb grafs aleatoris: 100 nodes, 500 arestes, 10 correspondències inicials.

Com podem veure a la Figura 6.12, introduint mig punt de soroll el graf obtingut amb *FIFO* aconsegueix reidentificar correctament prop del 35% dels nodes mentre que amb el mateix soroll, el percentatge de reidentificació augmenta fins al 95% en el graf obtingut amb *greedy*.

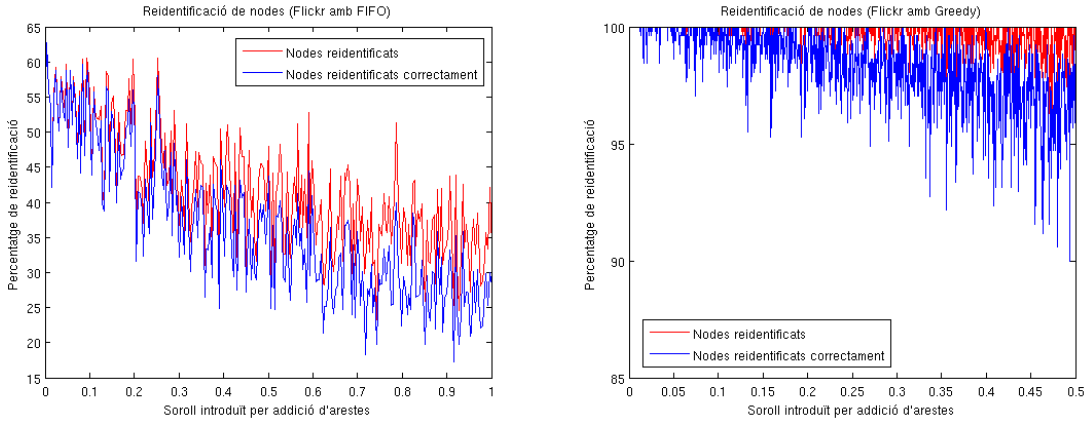

Figura 6.12: Agregació de grafs socials: grafs obtinguts amb *FIFO* (esquerra) i *greedy* (dreta), coeficients d'agrupaments mitjans de 0.343 i 0.636, respectivament.

6.3.3 Simulació de l'agregació real

Les proves realitzades fins aquest punt no permeten analitzar el comportament de l'algorisme en les mateixes condicions amb les que es treballarà amb l'agregació amb dades reals. Aquest tercer conjunt de proves intenta aproximar-se al problema recreant unes condicions similars a les que es trobaran en l'agregació real.

Per aquestes proves, es parteix d'un graf $G = (V, E)$, del qual se'n mostregen dos subconjunts de nodes $V_1 \in V$ i $V_2 \in V$ de manera que V_1 i V_2 presenten α_V nodes comuns[1](veure Figura 6.13). Posteriorment, es creen dues còpies d'E (E_1 i E_2) i s'eliminen una fracció α_E d'arestes de cada una de les còpies independentment. Per últim, es projecten els conjunts d'arestes modificades (E_1 i E_2) sobre els conjunts de nodes mostrejats (respectivament, V_1 i V_2) i es realitza l'agregació dels grafs obtinguts amb aquesta projecció (G_1 i G_2).

Podem veure com, en aquest cas, l'escenari és similar a una agregació real: disposem de dos grafs que tenen un conjunt de nodes comú però on les arestes que uneixen aquests nodes no sempre coincideixen. Variant el percentatge de nodes comuns entre V_1 i V_2 (α_V) podrem analitzar el comportament de l'algorisme d'agregació per diferents nivells de superposició. De la mateixa manera, variant el valor de la fracció d'arestes eliminades (α_E) en la creació d'E_1 i E_2, podem mesurar els efectes de la introducció de soroll en el procés d'agregació.

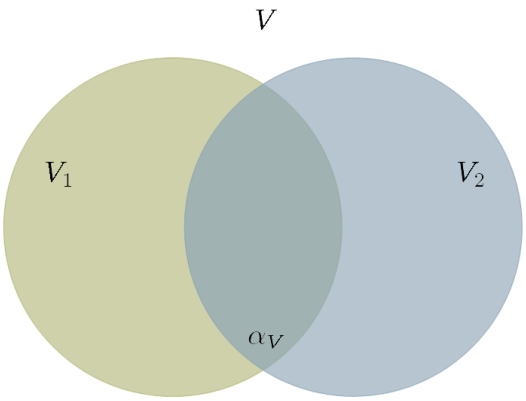

Figura 6.13: Diagrama de Venn dels conjunts de nodes en una agregació real.

A la Figura 6.14 hi podem veure el resultat d'aplicar el procediment descrit per un graf G de 200 nodes amb una superposició α_V d'1/3. Els dos subgrafs generats G_1 i G_2 tenen 134 nodes cada un, dels quals 68 són comuns. En aquest cas, en comptes de parlar de soroll farem servir una mesura inversa, la fracció d'arestes comuns (α_E). Podem veure com l'algorisme aconsegueix reidentificar correctament tots els nodes comuns[2] quan les arestes entre els nodes comuns de G_1 i G_2 coincideixen i com decau el percentatge de reidentificació conforme α_E decreix. També podem apreciar com l'algorisme comet errors de reidentificació, donant per vàlides correspondències que no existeixen. Aquests errors es poden disminuir augmentant el llindar d'excentricitat exigida per establir una nova correspondència però fer-ho comporta que decaigui també el número de nodes reidentificats correctament.

[1]Es mesura la superposició en termes del coeficient de Jaccard, que defineix la superposició entre dos conjunts X i Y com a $JC(X,Y) = \frac{\|X \cap Y\|}{\|X \cup Y\|}$ sempre que $X \cup Y \neq \emptyset$

[2]Si la superposició entre V_1 i V_2 és d'1/3, aleshores el màxim número de nodes que es podran reidentificar és el 33.3% del total de nodes

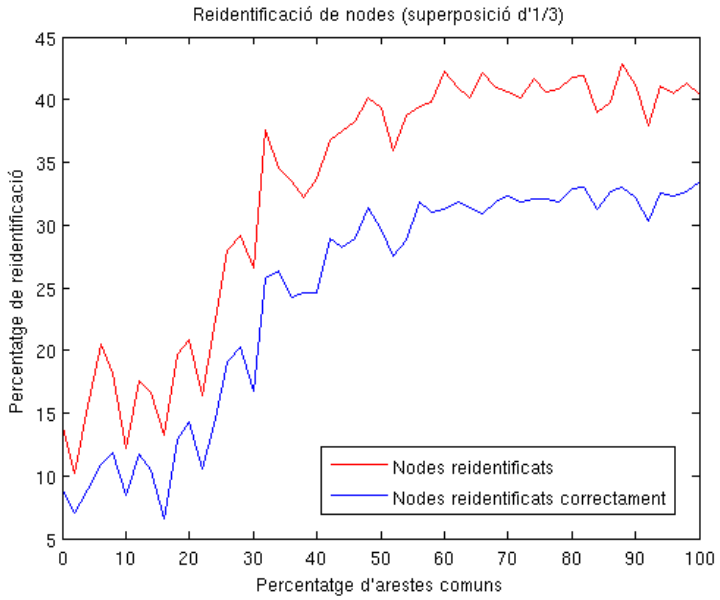

Figura 6.14: Simulació de l'agregació real: graf complet de 200 nodes amb superposició d'1/3 dels nodes.

6.3.4 Agregació amb dades reals

L'últim conjunt de proves s'ha realitzat agregant els grafs obtinguts de les diferents xarxes socials. Les correspondències inicials s'han establert a partir de les coincidències entre els noms d'usuari de les diferents xarxes. Com que no es disposa dels resultats correctes de les correspondències entre els nodes de les diferents xarxes socials, l'avaluació dels resultats es basarà en la comprovació manual de la sortida que ofereixi l'algorisme.

Inicialment, s'ha intentat realitzar l'agregació dels grafs esmentats a la Taula 6.1 però els resultats obtinguts no han estat satisfactoris. El número de correspondències inicials trobades a partir del nom d'usuari és molt baix (en el millor dels casos, de 3), número insuficient per a realitzar l'agregació correctament. Analitzant els usuaris que formen part dels diferents grafs manualment, s'ha trobat que el percentatge de superposició entre aquests era també baix[4], dificultant per tant, la tasca d'agregació. Així, per exemple, els grafs explorats amb *FIFO* de les xarxes *Twitter* i *Flickr* només presenten 2 usuaris amb nom coincident i una superposició d'aproximadament 8 usuaris. És evident, per tant, que els grafs explorats són massa petits per poder dur a terme el procés d'agregació. En les proves realitzades agregant aquests grafs, s'aconsegueix reidentificar correctament 1 sol node a partir de les correspondències inicials trobades amb noms d'usuari coincidents.

Amb l'objectiu de millorar els resultats anteriors s'ha repetit l'experiment augmentant la

[4]La superposició és baixa per aconseguir realitzar l'agregació amb grafs d'aquesta mida. Tot i així, una superposició del 8% en grafs grans és suficient per aconseguir l'agregació tal i com demostren els experiments realitzats a [35]

mida dels grafs de dues maneres diferents. Per una banda, s'han utilitzat els grafs descoberts en comptes dels explorats i, per l'altra, s'han realitzat *crawlings* amb més número de nodes explorats.

El segon intent d'agregació s'ha realitzat fent servir els grafs descoberts, és a dir, els grafs que contenen tots els usuaris descoberts, hagin estat o no explorats. Els nodes descoberts que no han estat explorats serviran per obtenir més correspondències inicials amb les quals començar l'algorisme, però, com hem vist anteriorment, difícilment crearan noves correspondències en la fase de propagació. Fent servir els grafs descoberts s'aconsegueix augmentar notablement el número de correspondències inicials. Així, per exemple, s'arriba a 66 usuaris amb noms coincidents agregant els grafs obtinguts amb *FIFO* de les xarxes *Lastfm* i *Flickr*. Tot i això, l'algorisme no aconsegueix propagar aquestes llavors i s'aconsegueix només 2 noves correspondències correctes. El motiu pel qual, tot i que l'algorisme disposa de 66 llavors, no és capaç de propagar-les el trobem en la distribució d'aquestes llavors. La majoria de les llavors (58) són nodes no explorats que estan poc connectats a la resta del graf. A més, els nodes explorats tenen graus elevats i, per tant, és necessari tenir força correspondències entre els nodes veïns per tal d'establir-ne una de nova (ja que, com hem vist, es divideix la puntuació de cada node per l'arrel quadrada del seu grau).

Després d'observar els resultats anteriors, es va decidir explorar una quantitat de nodes més elevada per tal d'intentar millorar el procés d'agregació. Els resultats però, tampoc no van millorar notablement. En les proves realitzades agregant un graf obtingut de la xarxa *Lastfm* amb 1200 nodes explorats amb un graf de la xarxa *Flickr* amb 600 nodes explorats, s'aconsegueixen establir 21 coincidències de noms d'usuari. Aquestes coincidències segueixen sense ser suficients per propagar-se i establir noves correspondències.

CAPÍTOL 7

Conclusió

7.1 El desenvolupament del projecte

Aquest projecte presenta una introducció força completa als diferents aspectes de l'anàlisi de xarxes socials: començant pels mecanismes de recol·lecció de dades fins als efectes sobre la privacitat dels usuaris, s'han tractat temes com les propietats dels grafs socials, les eines utilitzades per analitzar-los o les seves utilitats pràctiques per a diferents sectors.

Els objectius principals del projecte han estat assolits: s'han analitzat les propietats que presenten els grafs socials, s'ha recol·lectat informació de diferents xarxes socials *online* i se n'ha extret una part del seu graf social i s'ha realitzat el procés d'agregació entre els diferents grafs obtinguts. Tot i això, no tots els subobjectius plantejats inicialment s'han completat exitosament. Inicialment, ens plantejàvem extreure informació de la xarxa social *Facebook*, objectiu que va ser abandonat al veure els problemes que comportava. També en un principi es volia obtenir alguna mesura per quantificar la millora global de la informació obtinguda respecte a cada una de les fonts utilitzades. Aquest objectiu no s'ha realitzat per les restriccions temporals del projecte. En quant al procés d'agregació, hem pogut veure com s'ha implementat i provat amb èxit l'algorisme d'agregació en diversos escenaris. Tot i així, els resultats obtinguts en l'agregació dels grafs reals no han estat del tot satisfactoris. Per altra banda, s'han aconseguit assolir objectius que no s'havien plantejat inicialment i que han anat sorgint durant el desenvolupament del projecte. Així, per exemple, s'han implementat diferents algorismes de planificació i s'han estudiat els efectes que produeixen sobre el graf obtingut, s'ha desenvolupat una heurística que permet detectar els enllaços dels *blogrolls* i s'ha implementat un mecanisme per anonimitzar les exploracions a través de la xarxa *Tor*.

La planificació inicial del projecte ha estat força acurada, patint variacions considerables només en la fase d'experimentació amb les dades recollides (tasca 4 de la planificació inicial). Aquesta tasca s'ha allargat molt més del que estava previst ja que tant la realització de totes les proves preliminars (per obtenir els valors de configuració dels paràmetres de l'algorisme) com les primeres proves d'agregació reals han comportat més dedicació de l'esperada. La resta de tasques s'han acomplert amb variacions de com a molt una setmana respecte el que estava planificat.

7.2 Els resultats obtinguts

Una de les primeres dades que sorprenen a l'analitzar els grafs obtinguts és l'elevat grau que presenten els nodes explorats. Tot i que els resultats obtinguts són sobreestimacions del grau real de la xarxa, aquests valors són molt elevats. Comparant aquests valors amb les dades obtingudes de les mateixes xarxes socials al 2007[34], ens podem fer una idea del gran creixement que estan tenint les xarxes socials *online* ens els últims anys. Així, per exemple, mentre el grau mitjà de *Twitter* al 2007 era de 37.7, el graf obtingut amb menor grau de la xarxa *Twitter* amb el nostre *crawler* presenta un grau mitjà de 2317.7, més de 60 vegades el valor obtingut al 2007. També notable, tot i que no tant espectacular, és el creixement que ha tingut la xarxa *Flickr*, que presentava un grau mitjà de 32.2 al 2007 i de 336.6 en els nostres grafs, multiplicant per 10 el valor en només 3 anys.

També hem pogut observar les grans diferències entre els grafs obtinguts amb els diversos algorismes de planificació. Aquestes diferències ens mostren com les dades obtingudes per un procés de *crawling* tenen un biaix introduït per l'algorisme de planificació utilitzat. Aquest biaix haurà de ser tingut en compte en posteriors estudis a l'hora d'analitzar les dades obtingudes.

Un altre dels fets que hem pogut observar analitzant els resultats obtinguts és que els *blogs* són la xarxa social amb característiques més diferenciades de la resta de xarxes analitzades. Presenten un grau mitjà molt inferior a les altres xarxes, un coeficient d'agrupament mitjà també menor i petites diferències en els resultats dels diferents algorismes de planificació. El grau mitjà petit és un resultat evident ja que els *blogrolls* no acostumen a tenir més de 10 enllaços i, en canvi, els usuaris de les OSN tendeixen a tenir un nombre de relacions molt elevat. Les altres alteracions són produïdes, principalment, pel fet que els *blogrolls* presenten algunes irregularitats a l'hora de ser considerats una xarxa social. Mentre que a la resta de xarxes analitzades, cada node correspon normalment a un sol usuari, en els *blogrolls* aquesta correspondència no es tan directa: hi ha nodes que corresponen a més d'un usuari (*blogs* amb múltiples autors), usuaris amb més d'un node (autors amb més d'un *blog*) i nodes que no tenen cap correspondència directa amb cap usuari concret (per exemple, *blogs* corporatius o enllaços que, tot i trobar-se al *blogroll*, no són *blogs*). Per tots aquests motius, els grafs socials creats de la interacció entre diferents *blogs* presenten característiques diferents a la resta de

grafs analitzats.

Respecte al procés d'agregació, hem pogut veure com l'algorisme és efectiu si els grafs a agregar són prou grans, presenten un coeficient d'agrupament alt, presenten un mínim de nodes comuns i es poden establir suficients correspondències inicials entre aquests nodes. En canvi, hem vist com obtenint grafs petits amb poca superposició l'algorisme d'agregació no és capaç de propagar les correspondències inicials i aporta poques noves correspondències.

7.3 La privacitat de les connexions dels usuaris d'una xarxa social

En els últims mesos hi ha hagut grans polèmiques sobre la protecció de la privacitat dels usuaris. Així, mentre que *Google* ha reconegut recopil·lar informació sobre la localització dels *routers wifi* ([10]) que detectava mentre realitzava fotografies per l'*Street View*, *Facebook* canviava una vegada més els paràmetres de privacitat configurats per defecte en tots els perfils d'usuari per tal de fer-los més públics. Pocs mesos abans (al gener), Mark Zuckerberg se sumava a la llista de CEOs ([46], [26], [44]) que declaraven el final de l'era de la privacitat, provocant dures crítiques per part de la majoria d'experts alhora que noves discussions sobre el que es considerava avui en dia privacitat ([41], [11]). Mentre s'escrivien aquestes línies, *Facebook* tornava a anunciar un nou canvi en les configuracions de privacitat dels usuaris ([9]) per les crítiques rebudes per la modificació anterior, assegurant que simplificarien la interfície per tal d'ajudar als usuaris a ajustar les polítiques al que ells esperen del servei.

La realitat és que les dades tenen un gran valor per qui les controla i, per tant, moltes companyies fan el possible per aconseguir-les i controlar-les, fent que els usuaris en perdin el control. En aquest sentit, l'aparició de les xarxes socials *online* contribueix a aquesta pèrdua de control sobre les pròpies dades dels usuaris que les fan servir. Per una banda, la falta de comprensió dels esquemes de compartició de les xarxes socials *online* provoca que molts usuaris facin pública informació creient que l'estan compartint únicament amb els seus amics o familiars. Per altra banda, els usuaris no acostumen a ser conscients d'aquesta nova font d'informació que ofereixen les xarxes socials: les connexions entre els usuaris. Aquesta falta de coneixement se suma també a la dificultat de controlar la informació continguda en les relacions, que sovint surt del control del propi usuari. Les relacions entre els usuaris suposen, a part d'informació directa sobre els mateixos, una font per establir correspondències entre diferents conjunts de dades, arribant a permetre desanonimitzar-los i comprometent així la privacitat dels usuaris que hi apareixen.

El fet que el procés d'agregació de diferents conjunts de dades sigui viable fent servir únicament l'estructura dels grafs[1] demostra que l'anonimat no és suficient per garantir la privacitat dels usuaris d'una xarxa social. La publicació de grafs sense informació dels identificadors dels nodes no garanteix que no es pugui realitzar un procés de reidentificació, aconseguint el graf original amb un alt percentatge de nodes reidentificats. És evident, per tant, que cal dissenyar

altres mecanismes per poder mantenir la privacitat dels usuaris de les xarxes socials abans de publicar o cedir a tercers els grafs socials d'aquestes. La viabilitat del procés d'agregació a partir de l'estructura dels grafs també implica que no té massa sentit seguir parlant d'informació personal d'identificació (*PII, Personally Identificable Information*), és a dir, de dades que permeten identificar a un persona de forma única. Quin sentit té parlar de dades que identifiquen a una persona de forma única quan aquesta identificació es possible sense necessitat de cap dada concreta associada a la persona?

Hem afirmat que és necessari que les empreses que disposen de grans quantitats de dades en forma de grafs socials es preocupin de modificar-los convenientment abans de publicar-los per tal de garantir la privacitat dels seus usuaris, però hem deixat de banda què poden fer els usuaris que formen part d'aquests grafs per tal de conservar aquesta privacitat. Després d'analitzar en quins casos era més difícil realitzar el procés d'agregació, podem veure com mantenir poques relacions amb altres usuaris (tenir un grau baix) i que aquests estiguin poc relacionats entre ells (presentar un coeficient d'agrupament baix) és una de les millors alternatives per evitar ser víctimes de processos de reidentificació basats en l'agregació de grafs. Tot i això, observant els graus mitjans obtinguts de les exploracions realitzades i comparant-los amb les dades de les mateixes OSN obtingudes l'any 2007, podem veure a quina velocitat estan creixent les xarxes socials i fer-nos una idea de fins a quin punt pot ser difícil mantenir un grup reduït de relacions en una xarxa social *online*. Aquest augment del número de relacions unit a l'augment del número d'usuaris i de la quantitat d'informació que s'hi comparteix permet facilitar el procés d'agregació.

7.4 Treball futur

El projecte deixa obertes noves portes per a la investigació i millora del treball realitzat:

- *Millores del crawler:* Com hem vist, és necessari obtenir grafs de certa mida per tal de poder realitzar el procés d'agregació. Per tal d'aconseguir grafs grans de manera més ràpida, probablement sigui necessari canviar la política d'intentar no fer un ús intensiu de la xarxa a explorar per una política una mica més agressiva, que intentés maximitzar el nombre d'usuaris explorats per unitat de temps.

- *Algorismes de planificació:* Les diferències que presenten els grafs obtinguts amb els diferents algorismes de planificació del *crawler* permeten fer-nos una idea del biaix introduït en les dades. Estudiar aquest biaix detingudament permetria poder extrapolar més acuradament els resultats obtinguts de l'anàlisi d'un subgraf a la xarxa social completa.

[1]Els atributs associats als nodes es fan servir en aquest projecte per a la fase d'inicialització però altres mètodes alternatius són viables per realitzar aquesta fase

- *El graf social format pels blogs:* Analitzar el graf social obtingut dels *blogrolls* és també una línia d'investigació en la qual s'hi pot treballar àmpliament. El baix grau presentat pels nodes d'aquest graf social el fa una font d'informació valuosa ja que permet definir de manera més acurada el cercle d'amistats o interessos del propietari del *blog*. Altres grafs socials obtinguts de fonts que no són estrictament una OSN (per exemple, el graf social obtingut de relacionar persones que apareixen en les mateixes fotografies) són també altres possibles vies d'investigació.

- *Actualització del graf explorat:* Un altre dels temes interessants des del punt de vista d'obtenció dels grafs socials és la seva actualització. Una vegada s'ha obtingut un graf social, pot ser necessari tenir-lo actualitzat sense haver de tornar a explorar tot el graf. El disseny d'algorismes que permetin fer-ho eficientment és també un possible tema a tractar.

- *Experimentació amb grafs dirigits:* Tot i que tant el *crawler* com l'algorisme d'agregació implementats permeten tractar amb grafs dirigits, les proves realitzades s'han dut a terme únicament amb grafs simètrics, fent servir els grafs induïts quan les xarxes analitzades oferien grafs dirigits. Queda com a treball futur, per tant, realitzar l'experimentació amb grafs dirigits.

- *Millores en l'agregació:* Una de les propietats que hem vist dels grafs socials és la seva estructuració en comunitats. La identificació d'aquestes comunitats en els diferents grafs a agregar pot ser útil tant durant la fase d'inicialització de l'algorisme com una vegada finalitzada l'agregació. Analitzar aquesta i altres possibles millores de l'algorisme d'agregació són possibles vies de treball futur.

- *Identificació de comunitats:* Seguint amb la identificació de les comunitats dels grafs socials, un altre tema que seria interessant estudiar és la relació entre les comunitats detectades pels algorismes d'identificació de comunitats i les comunitats o grups que creen explícitament els usuaris a les xarxes socials *online*. Estudiar aquesta relació permetria, per una banda, millorar els algorismes d'identificació de comunitats i, per l'altra, entendre millor com s'organitzen les persones en comunitats.

- *Resistència a l'agregació:* Des del punt de vista de la privacitat dels usuaris, seria interessant estudiar alternatives que permetessin mantenir-la sense que impliquessin restriccions massa estrictes. Hem vist que mantenir un grau i un coeficient d'agrupament baixos dificulten el procés d'agregació però exigir als usuaris d'una OSN que compleixin aquestes condicions va en contra del propi ús de les OSN. Buscar alternatives que dificultin el procés d'agregació alhora que mantinguin la funcionalitat de les OSN és per tant, un problema obert.

Bibliografia

[1] A. Mohammad Abdulkader. *Parallel Algorithms for Labeled Graph Matching.* PhD thesis, Colorado School of Mines, 1998.

[2] David Auber and Patrick Mary. Tulip. `http://tulip.labri.fr`.

[3] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 181–190, New York, NY, USA, 2007. ACM.

[4] R. B. Baldwin. Lunar crater counts. 69:377–+, June 1964.

[5] Partha Basuchowdhuri. *Greedy methods for approximate graph matching with applications for social network analysis.* PhD thesis, Louisiana State University and Agricultural and Mechanical College, 2009.

[6] A. Bavelas. A mathematical model for group structures. *Human Organization*, (7):16–30, 1948.

[7] A. Bavelas. Communication patterns in task oriented groups. *Journal of the Acoustical Society of America*, (22):271–282, 1950.

[8] A. Bilgin, J. Ellson, E. Gansner, Y. Hu, Y. Koren, and S. North. Graphviz project. `http://www.graphviz.org/`.

[9] "The Official Google Blog". From facebook, answering privacy concerns with new settings. `http://almendron.com/tribuna/30104/from-facebook-answering-privacy-concerns-with-new-settings/`, 2010.

[10] "The Official Google Blog". Wifi data collection: An update. `http://googleblog.blogspot.com/2010/05/wifi-data-collection-update.html`, 2010.

[11] Danah Boyd. Making sense of privacy and publicity. `http://www.danah.org/papers/talks/2010/SXSW2010.html`, 2010.

[12] Ulrik Brandes and Dorothea Wagner. Visone - analysis and visualization of social networks. `http://visone.info/`, 2006.

[13] Dan Brickley and Libby Miller. FOAF vocabulary specification. `http://xmlns.com/foaf/spec/20050403.html`, 2005.

[14] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38, 2006.

[15] Heather Champ. 4,000,000,000. http://blog.flickr.net/en/2009/10/12/4000000000/, 2009.

[16] B. S. Cohn and M Marriot. Networks and centers of integration in indian civilization. *Journal of Social Research*, (1):1–9, 1958.

[17] Oracle Corporation. Java. http://www.java.com.

[18] Oracle Corporation. Mysql. http://www.mysql.com/.

[19] J. A. Czepiel. Word of mouth processes in the diffusion of a major technological innovation. *Journal of Marketing research*, (11):172–180, 1974.

[20] Roger Dingledine and *et al.* Tor project. http://www.torproject.org/.

[21] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, volume 29, pages 251–262, New York, NY, USA, October 1999. ACM.

[22] Joe Foley. Torlib. http://www.mit.edu/~foley/TinFoil/Docs/tinfoil/TorLib.html.

[23] Beno Gutenberg and Charles Richter. *Seismicity of the Earth and associated phenomena.* Princeton University Press, Princeton, New Jersey, 2nd edition, 1954.

[24] S.L. Hakimi. Optimum locations of switching centers and the absolute centers and medians graph. *Operations Research*, (12):450–459, 1965.

[25] Paul W. Holland and Samuel Leinhardt. Transitivity in structural models of small groups. page 107–124, 1971.

[26] Bobbie Johnson. Privacy no longer a social norm, says facebook founder. http://www.guardian.co.uk/technology/2010/jan/11/facebook-privacy, 2010.

[27] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, April 1989.

[28] Martijn Koster. A standard for robot exclusion. http://www.robotstxt.org/norobots-rfc.txt, 1996.

[29] Sang H. Lee, Pan-Jun Kim, and Hawoong Jeong. Statistical properties of sampled networks. Nov 2009.

[30] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, New York, NY, USA, 2006. ACM Press.

[31] Kenneth D. Mackenzie. Structural centrality in communications networks. *Psychometrika*, (31):17–25, 1966.

[32] MathWorks. Matlab. http://www.mathworks.com/products/matlab/.

[33] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42, New York, NY, USA, 2007. ACM.

[34] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42, New York, NY, USA, 2007. ACM.

[35] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *SP '09: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pages 173–187, Washington DC, USA, Mar 2009. IEEE Computer Society.

[36] J. Nieminen. On the centrality in a directed graph. *Social Science Research*, (2):371–378, 1973.

[37] Stephen C. North. Drawing graphs with neato. http://www.graphviz.org/pdf/neatoguide.pdf, 2004.

[38] S. Redner. How popular is your paper? an empirical study of the citation distribution, April 1998.

[39] David L. Rogers. Sociometric analysis of interorganizational relations: application of theory and measurement. *Rural Sociology*, (39):487–503, 1974.

[40] G. Sabidussi. The centrality index of a graph. *Psychometrika*, (31):581–603, 1966.

[41] Bruce Schneier. Privacy and control.
`http://www.schneier.com/blog/archives/2010/04/privacy_and_con.html`, 2010.

[42] M. E. Shaw. Group structure and the behavior of individuals in small groups. *Journal of Psychology*, (38):139–149, 1954.

[43] A. Shimbel. Structural parameters of communication networks. *Bulletin of Mathematical Biophysics*, (15):501–507, 1953.

[44] Polly Sprenger. Sun on privacy: Get over it. `http://www.wired.com/politics/law/news/1999/01/17538`, 1999.

[45] Michael P. H. Stumpf, Carsten Wiuf, and Robert M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4221–4224, March 2005.

[46] Ryan Tate. Google ceo: Secrets are for filthy people.
`http://gawker.com/5419271/google-ceo-secrets-are-for-filthy-people`, 2010.

[47] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[48] W3C. FOAF sites. `http://esw.w3.org/FoafSites`, 2009.

[49] Yan Wang, Jui-Hung Hung, Yi-Chien Chan, Chia-Ling Huang, and Matt Huyck. Visant - interactive visual analysis tool for biological networks and pathways. `http://visant.bu.edu/`.

[50] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[51] Shaozhi Ye, Juan Lang, and Felix Wu. Crawling online social graphs. In *Proceedings of the 12th International Asia-Pacific Web Conference*, April 2010.

[52] G. K. Zipf. Selective studies and the principle of relative frequency in language, 1932.

Annexos

ANNEX A

L'algorisme d'agregació

```
% Eccentricity function: calculates eccentricity for a list of scores
% Params:
%   items: list of numerical values
% Returns:
%   ecc: eccentricity

function [ecc] = eccentricity (items)

std_dev = std(items);
[max1, index] = max(items);
items(index) = nan;
max2 = max(items);

if( std_dev ~= 0 )
    ecc = (max1 - max2) / std_dev;
else
    ecc = 0;
end
```

```
% Propagation step function: implements the agregation algorithm for two graphs
% Params:
%   lgraph: one graph
%   rgraph: another graph (the algorithm doesn't make differences between l and r graphs)
%   mapping: initial mapping between nodes of lgraph and rgraph
%   theta: eccentricity threshold (tradeoff between yield and accuracy)
%   thetaDec: factor which reduces theta at each iteration
%   maxIter: maximum number of iterations allowed
% Returns:
%   mapping: final mapping

function mapping = propagationStep( lgraph, rgraph, mapping, theta, thetaDec, maxIter )

if ( size(mapping) == [0,0] )
    return;
end
lastSize = 0;
eccFail = 1;
sizel = size(lgraph,1);
while ( ( size(mapping,1) ~= lastSize ) || ( eccFail) ) && ( maxIter ~= 0)
    disp('New iteration');
    lastSize = size(mapping,1);
    eccFail = 0;
    for lnode = 1 : sizel
        if isempty ( find(mapping(:,1) == lnode ) )
            % Compute scores for lnode
            lScores = matchScores( lgraph, rgraph, mapping, lnode);
            if( eccentricity(lScores) < theta )
                disp('lscores: Eccentricity < theta');
                eccFail = 1;
                continue;
            end;
            [value, rnode]  = max( lScores );

            % Compute scores for rnode
            rScores = matchScores( rgraph, lgraph, invert(mapping), rnode);
            if( eccentricity(rScores) < theta )
                disp('rScores: Eccentricity < theta');
                eccFail = 1;
                continue;
            end;
            [value, reverseMatch] = max( rScores );

            % Check for reverse match
            if( reverseMatch ~= lnode )
                disp('No reverse match');
                continue;
            end;
            disp('New mapping found');
            mapping = [ mapping ; lnode, rnode ];
        else
            disp('Already mapped');
        end
    end
    maxIter = maxIter - 1;
    theta = theta * thetaDec;
end
```

```
% Match scores function: calculates scores for a node
% Params:
%   lgraph: one graph
%   rgraph: another graph
%   mapping: current mapping between nodes of lgraph and rgraph
%   lnode: node from lgraph being evaluated
% Returns:
%   scores: scores for lnode

function [ scores ] = matchScores( lgraph, rgraph, mapping, lnode )

scores = zeros( 1, size(rgraph,1) );

% In degree
lgraphEdges = find( lgraph(:,lnode) );
for lnbr = 1 : size( lgraphEdges,1 )
    map = find( mapping(:,1) == lgraphEdges(lnbr) , 1);
    if ( isempty( map ) )
        continue;
    end

    rnbr = mapping( map,2 );

    rgraphEdges = find ( rgraph(rnbr,:) );
    for rnode = 1 : size( rgraphEdges,2 )
        map = find ( mapping(:,2) == rgraphEdges(rnode) , 1);
        if not ( isempty ( map ) )
            continue;
        end

        scores(rgraphEdges(rnode)) = scores(rgraphEdges(rnode)) +
            ( 1 / ( sum( rgraph(:,rgraphEdges(rnode)) ) ^ (1/2) ) );
    end
end

% Out degree
lgraphEdges = find( lgraph(lnode,:) );
for lnbr = 1 : size( lgraphEdges,2 )
    map = find( mapping(:,1) == lgraphEdges(lnbr) , 1);
    if ( isempty( map ) )
        continue;
    end

    rnbr = mapping( map,2 );

    rgraphEdges = find ( rgraph(:,rnbr) );
    for rnode = 1 : size( rgraphEdges,1 )
        map = find ( mapping(:,2) == rgraphEdges(rnode) , 1);
        if not ( isempty ( map ) )
            continue;
        end

        scores(rgraphEdges(rnode)) = scores(rgraphEdges(rnode)) +
            ( 1 / ( sum( rgraph(rgraphEdges(rnode),:) ) ^ (1/2) ) );
    end
end
```

Firmat: Cristina Pérez Solà
Bellaterra, juny de 2010

Resum

Aquest projecte mostra com les connexions dels usuaris d'una xarxa social suposen un risc afegit per a la privacitat dels usuaris que hi formen part. Aquestes connexions ofereixen informació suficient per a poder dur a terme processos d'agregació d'informació entre diferents xarxes socials, permetent a un atacant millorar el seu coneixement inicial sobre les xarxes. El projecte és un recorregut per totes les fases necessàries per dur a terme aquest procés, des de la recollida de la informació fins a l'agregació de les dades obtingudes.

Resumen

Este proyecto muestra como las conexiones de los usuarios de una red social suponen un riego añadido para la privacidad de los usuarios que la forman. Estas conexiones ofrecen información suficiente para poder realizar procesos de agregación de información entre diferentes redes sociales, permitiendo a un atacante mejorar su conocimiento inicial sobre las redes. El proyecto es un recorrido por las diversas fases necesarias para realizar este proceso, desde la recogida de información hasta la agregación de los datos obtenidos.

Abstract

This project shows that link connections in social networks suppose a new risk for user's privacy. These links provide an attacker with enough information to make aggregation processes between different online social networks, improving the attackers initial knowledge of the network. This project is a journey towards this aggregation process, starting from data collection and ending in the aggregation itself.