

Progetto di Social Computing

Parata Loris (144338)
Arzon Francesco (142439)
Dal Fabbro Lorenzo (142300)
Galvan Matteo (142985)

Indice

1	Sottografo di Twitter	2
1.1	Introduzione	2
1.2	Costruzione del grafo	2
1.2.1	Download dei nodi	2
1.2.2	Creazione degli archi	2
1.2.3	Visualizzazione del grafo	3
1.3	Analisi del grafo completo	4
1.3.1	Proprietà dei grafi	4
1.3.2	Misure di centralità	4
1.3.3	Albero di copertura minimo	5
1.4	Analisi del sottografo dell'utente KevinRoitero	5
1.5	Smallworldness	5
1.6	Analisi delle correlazioni di Pearson e Kendall	5
1.7	Conclusioni	6

Capitolo 1

Sottografo di Twitter

1.1 Introduzione

Il primo progetto di Social Computing consiste nello studio della rete sociale di 5 utenti di Twitter. Lo studio è stato svolto mediante la costruzione di un grafo, rappresentante rete sociale, costituito dai 5 utenti principali e della loro relativa rete di contatti costituita dai loro follower, following e da rispettivi sottoinsiemi campionati in maniera random. In specifico abbiamo analizzato la relazione diretta di **follow** tra tutti i nodi del grafo ed i 5 profili scelti.

1.2 Costruzione del grafo

1.2.1 Download dei nodi

Il primo passo consiste nello scaricare tutti i followers attraverso la `api.followers` di Twitter dei cinque nodi principali:

- @Mizzaro
- @damiano10
- @Miccighel_
- @eglu81
- @KevinRoitero

json e compagnia bella con codice se serve

Successivamente abbiamo scelto 5 followers e 5 following randomicamente per ognuno dei 5 account. In seguito, da ognuno di essi sono stati scelti altri 10 account followers e 10 account following sempre in maniera casuale.

Infine, una volta ottenuti tutti gli account, abbiamo scaricato tutte le informazioni principali relative agli account mediante `api.get user`.

Per un totale di 3101 nodi.

1.2.2 Creazione degli archi

Successivamente abbiamo controllato l'esistenza di una relazione tra tutti gli account scaricati ed i 5 nodi principali con la funzione `api.show friendship`. Aggiungendo gli archi raffiguranti l'azione di follow al grafo.

Ottimizzazione archi

E' possibile rilevare tutti i nodi direttamente connessi ai 5 account andando a visualizzare direttamente i rispettivi followers, riducendo significativamente i costi in termine di richieste all'API. Ma per attinenza alla traccia abbiamo fatto un controllo completo per ogni nodo scaricato precedentemente.

1.2.3 Visualizzazione del grafo

La visualizzazione interattiva del grafo costruito con le funzioni messe a disposizione di networkX avviene utilizzando la libreria apposita pyvis.

Ottimizzazione visualizzazione

E' possibile ridurre i costi per l'elaborazione grafica di costruzione del grafo impostando il parametro opzionale `phisc = False`. Questo parametro a discapito dell'interazione fisica nel trascinamento dei nodi che avrebbero una risposta fisica, permette di risparmiare l'80 per cento di tempo.

1.3 Analisi del grafo completo

Applicando le relative funzioni messe a disposizione dalla libreria di networkX abbiamo potuto stabilire che il grafo è:

Il grafo da noi analizzato è risultato **non connesso**.

Questo sottolinea che è errato dar per scontato che tutti gli utenti che seguono un determinato account **UtenteTwitter** a loro volta sono seguiti da utenti che seguono anche loro l' **UtenteTwitter**.

Nel caso in cui tenessimo traccia delle relazioni interne tra i nodi di secondo livello e quelli di terzo livello, considerando i path indiretti, allora sarebbe risultato connesso.

Ma questo dipende dalla componente casuale che sceglie da quali nodi scaricare i relativi follower dei follower.

Ecco perchè abbiamo deciso di procedere alla rimozione dei nodi sconnessi del grafo principale e di confrontare le proprietà dei due grafi dove è possibile.

1.3.1 Proprietà dei grafi

Il grafo completo risulta **non connesso**, mentre il suo sottografo è **connesso**.

Ma entrambi risultano **non bipartiti**.

Centro, Diametro e Raggio sono calcolabili solamente per il sottografo perchè il grafo completo essendo sconnesso ha valore di diametro e raggio infinito per definizione.

Per il sottografo abbiamo :

- Centro: i nodi di **Kevin Roitero**, **Gianluca Demartino** e **Damiano Spina**, tre dei 5 nodi principali.
- Diametro: 4
- Raggio: 2

1.3.2 Misure di centralità

Questi valori sono calcolabili su entrambi i grafi e sono i seguenti

- Betweenness: il valore
- Closness:
- Degree:
- In-centrality:
- Out-centrality:
- Degree:

Riportiamo i valori relativi al sottografo connesso perché i risultati sono coerenti ai valori ottenuti dal grafo principale.

- PageRank: il nodo di Damiano Spina ha il valore più alto(0,243),seguito dai restanti nodi principali, in ordine di numero di followers.
- Hits:
 - Hubs: l'hub principale è Luke Gallagher, con valore 0.0015, che segue 3 profili su 5.
 - Authorities: la maggiore authority è Damiano con valore di 0.6, seguito dai restanti nodi principali

Questo conferma come le autorità siano i nodi con più archi entranti, in questo caso i 5 nodi principali, mentre gli hubs siano i nodi che seguono, in questo caso, più nodi principali.

1.3.3 Albero di copertura minimo

Gli archi tra i nodi che compongono l'albero di copertura minimo è possibile visualizzarli all'interno file *SC_progetto.ipynb*.

1.4 Analisi del sottografo dell'utente KevinRoitero

Considerando il sottografo dell'account KevinRoitero:

Lo estraiamo utilizzando la funzione `nx.ego_graph(sub_twitter_graph.reverse(), "3036907250")` sul sottografo, ma considerando la sua versione reverse, essendo che siamo interessati ai nodi che seguono Kevin (id: 3036907250). Vista la complessità computazionale dell'operazione di maxClique, l'abbiamo applicata su uno dei sottografi più piccoli del grafo principale.

1.5 Smallworldness

Considerando il sottografo non direzionato il valore dei parametri:

- Omega: 0.00893
- Sigma:

Il valore di omega vicino allo 0, indica che il grafo ha le caratteristiche di una rete piccolo mondo. I parametri sono stati calcolati con:

- niter = 10, Numero approssimativo di ridirezioni per arco, per calcolare il grafico casuale equivalente

- nrand = 2, Numero di grafici casuali generati per calcolare il coefficiente di clustering medio e la lunghezza del percorso medio più breve.

Abbiamo optato per valori più bassi rispetto a quelli di default (100,10), questo per ridurre la complessità computazionale del calcolo dei parametri.

1.6 Analisi delle correlazioni di Pearson e Kendall

1.7 Conclusioni

Dopo quest'analisi possiamo affermare che il grafo sociale costruito evidenzia la centralità dei nodi principali nel grado, ma possiamo anche notare la presenza di nodi completamente scollegati da esso. I risultati ottenuti dalle funzioni ci permette di affermare che
Se ci fosse stata la possibilità di costruire l'intera rete sociale dei 5 nodi principali, con delle API di Twitter meno restringenti, avremmo potuto notare l'eventuale presenza di altri Hub o autorità.