

# Progetto di Social Computing

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

# Indice

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Sottografo di Twitter</b>                              | <b>2</b> |
| 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 . . . . . | 6        |
| 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 3103 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: come valore massimo abbiamo 0.000375 riferito al nodo di Damiano, seguito da eglu e micc. Si presenta un valore coerente anche nel sottografo, ma con valori più elevati,(0.0012)
- Closness: come valore massimo 0.353 riferito al nodo di Damiano, seguito da... Si presenta un valore coerente anche nel sottografo, ma con valori più elevati,(0.653)
- Degree: come valore massimo 0.254 riferito al nodo di Damiano, seguito da... Si presenta un valore coerente anche nel sottografo, ma con valori più elevati,(0.653)
- In-centrality: 0.253, 0.469
- Out-centrality: 0.00128 di Kevin Callegher, seguito da nodi che seguono 3 profili main e poi 2. ... 0.00238 sub

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: 0.00683
- Sigma: 0.98653

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

|    |                |                |   |   |
|----|----------------|----------------|---|---|
| 0  | btw centrality | cls centrality | (0.8733993223251802, 0.0)                     | (1.0, 0.0)                                    |
| 1  | btw centrality | dg centrality  | (0.9910425083190649, 0.0)                     | (0.07646960324418119, 1.2850857755873085e-05) |
| 2  | btw centrality | in centrality  | (0.9916571758957484, 0.0)                     | (1.0, 0.0)                                    |
| 3  | btw centrality | out centrality | (0.12736453492501887, 1.071886184584708e-12)  | (0.07449925773728967, 2.137232727948775e-05)  |
| 4  | cls centrality | btw centrality | (0.8733993223251802, 0.0)                     | (1.0, 0.0)                                    |
| 5  | cls centrality | dg centrality  | (0.9041405908224469, 0.0)                     | (0.07646960324418119, 1.2850857755873085e-05) |
| 6  | cls centrality | in centrality  | (0.9033962532946789, 0.0)                     | (1.0, 0.0)                                    |
| 7  | cls centrality | out centrality | (0.15199990321507398, 1.6921549195351964e-17) | (0.07449925773728967, 2.137232727948775e-05)  |
| 8  | dg centrality  | btw centrality | (0.9910425083190649, 0.0)                     | (0.07646960324418119, 1.2850857755873085e-05) |
| 9  | dg centrality  | cls centrality | (0.9041405908224469, 0.0)                     | (0.07646960324418119, 1.2850857755873085e-05) |
| 10 | dg centrality  | in centrality  | (0.9993531356368165, 0.0)                     | (0.07646960324418119, 1.2850857755873085e-05) |
| 11 | dg centrality  | out centrality | (0.16327635144167213, 5.53423015238216e-20)   | (0.9998902123524452, 0.0)                     |
| 12 | in centrality  | btw centrality | (0.9916571758957484, 0.0)                     | (1.0, 0.0)                                    |
| 13 | in centrality  | cls centrality | (0.9033962532946789, 0.0)                     | (1.0, 0.0)                                    |
| 14 | in centrality  | dg centrality  | (0.9993531356368165, 0.0)                     | (0.07646960324418119, 1.2850857755873085e-05) |
| 15 | in centrality  | out centrality | (0.12769071521714853, 9.37845594949372e-13)   | (0.07449925773728967, 2.137232727948775e-05)  |
| 16 | out centrality | btw centrality | (0.12736453492501887, 1.071886184584708e-12)  | (0.07449925773728967, 2.137232727948775e-05)  |
| 17 | out centrality | cls centrality | (0.15199990321507398, 1.6921549195351964e-17) | (0.07449925773728967, 2.137232727948775e-05)  |
| 18 | out centrality | dg centrality  | (0.16327635144167213, 5.53423015238216e-20)   | (0.9998902123524452, 0.0)                     |
| 19 | out centrality | in centrality  | (0.12769071521714853, 9.37845594949372e-13)   | (0.07449925773728967, 2.137232727948775e-05)  |

Figura 1.1:

## 1.7 Conclusioni

Dopo quest'analisi possiamo affermare che il grafo sociale costruito evidenzia la centralità dei nodi principali nel grafo, ma possiamo anche notare la presenza di nodi completamente scollegati da esso. I risultati ottenuti dalle funzioni ci permette di affermare che il sottografo connesso si tratta di una rete **"Piccolo Mondo"**.