Università Politecnica delle Marche

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica e dell'Automazione

Social Network Analysis su una rete di autori accademici

Docenti
Prof. Ursino Domenico
Prof. Luca Virgili

Studenti
Mori Nicola
Sospetti Mattia
Zitoli Francesca

Anno accademico 2021-2022

Indice

1 Introduzione

Con questo elaborato si vuole perseguire l'obbiettivo di effettuare una analisi su una particolare rete sociale. La Social Network Analysis (SNA) può essere definita come lo studio delle relazioni umane per mezzo della teoria di grafi. Essa viene usata molto nei social media in quanto i dati sono reperibili (basti pensare a Reddit o in parte anche Twitter), e perché comunque apre alla possibilità di approfondimenti assolutamente interessanti dal punto di vista informativo ed anche lucrativo. Per effettuare questo studio è stato usato un tool di Python, che prende il nome di NetworkX: tale libreria permette l'analisi di una rete (e così anche di un grafo), attraverso una lunga serie di funzionalità offerte.

1.1 NetworkX

NetworkX (Figura 1) è un pacchetto open source per il linguaggio Python per la creazione, manipolazione e analisi di strutture, dinamiche e funzioni di reti complesse. Esso permette di costruire delle strutture dati per grafi semplici, direzionali e multigrafi, fornendo gli algoritmi standard per l'analisi delle caratteristiche e delle misure di grafi. Presenta inoltre una notevole flessibilità nella rappresentazione e gestione dei grafi consentendo di utilizzare etichette di archi e nodi di qualsiasi forma (testi, immagini, record XML, ecc).

Figura 1: Logo di NetworkX

1.2 Dataset

Lo studio di questa sezione inizia con la presentazione del dataset su cui si è incentrata l'analisi. Il dataset in questione, reperibile sul sito dell'Università di Stanford al link `https://snap.stanford.edu/data/ca-GrQc.html`, considera le collaborazioni scientifiche tra gli autori degli articoli inviati alla categoria "Relatività Generale e Cosmologia Quantistica". La logica secondo cui è stato costruito il grafo è la seguente: se un autore i è stato coautore di un articolo con l'autore j, allora nel grafo comparirà un arco indiretto che collega i due nodi i e j. Per completezza, riportiamo che il dataset tiene in

considerazione un arco temporale importante: 124 mesi (da Gennaio 1993 a Aprile 2003), il che ha dato alla luce un dataset di dimensioni importanti (5242 nodi e 14496 archi), che, come vedremo nella sezione dedicata, ha creato non pochi problemi, risolti successivamente con una oculata fase di ETL.

Per chi fosse interessato, di seguito riportiamo altri dati relativi al nostro dataset (Figura 2):

Dataset statistics	
Nodes	5242
Edges	14496
Nodes in largest WCC	4158 (0.793)
Edges in largest WCC	13428 (0.926)
Nodes in largest SCC	4158 (0.793)
Edges in largest SCC	13428 (0.926)
Average clustering coefficient	0.5296
Number of triangles	48260
Fraction of closed triangles	0.3619
Diameter (longest shortest path)	17
90-percentile effective diameter	7.6

Figura 2: Dati strutturali

1.2.1 ETL

Come anticipato nella fase di presentazione del dataset, è stato necessaria, al fine di rendere più scorrevole tutta la fase di analisi ed anche quindi la futura illustrazione dei risultati ottenuti, una preliminare fase di ETL (Extract/Transform/Load).

La prima operazione di ETL è stata pressocché obbligatoria: sono state rimosse le prime 4 righe puramente testuali che descrivevano la logica del dataset.

Di seguito ci siamo resi subito conto che sul file originale gli indici dei nodi venivano separati con un particolare carattere (che avrebbe dato poi in futuro problemi), a tal fine è stato rimpiazzato con un semplice carattere di spazio, come mostra la seguente figura (Figura 3):

Nonostante queste operazioni, in realtà abbiamo notato che ciò che veniva fuori era un grafo non connesso, ovvero mostrava più di una componente connessa: a tal proposito dunque si è deciso di filtrare il dataset attraverso una serie di istruzioni prese direttamente dalla libreria NetworkX di Python.

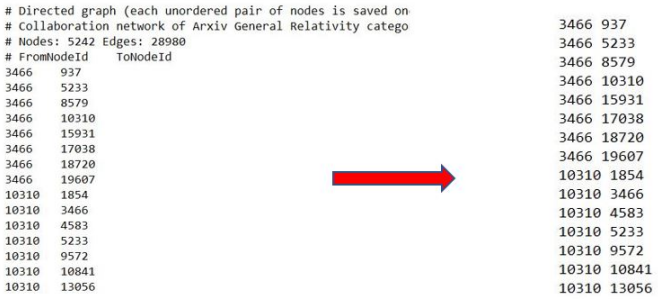


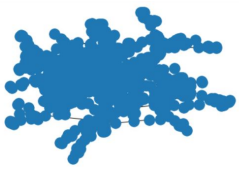
Figura 3: ETL

Tale sequenza di istruzioni inizialmente cattura la componente connessa di dimensioni (intese come numero di nodi appartenenti ad essa) maggiori, e costruisce un sottografo, considerando solo i nodi selezionati dall'istruzione precedente.

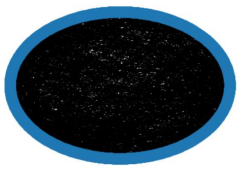
Dunque la rete dopo il filtro appare ragionevolmente di dimensioni inferiori: 784 nodi e 1514 archi, un numero che abbiamo considerato come un buon compromesso tra rappresentatività del dataset e possibilità di farci analisi in modo agevole.

2 Layout e Struttura della rete

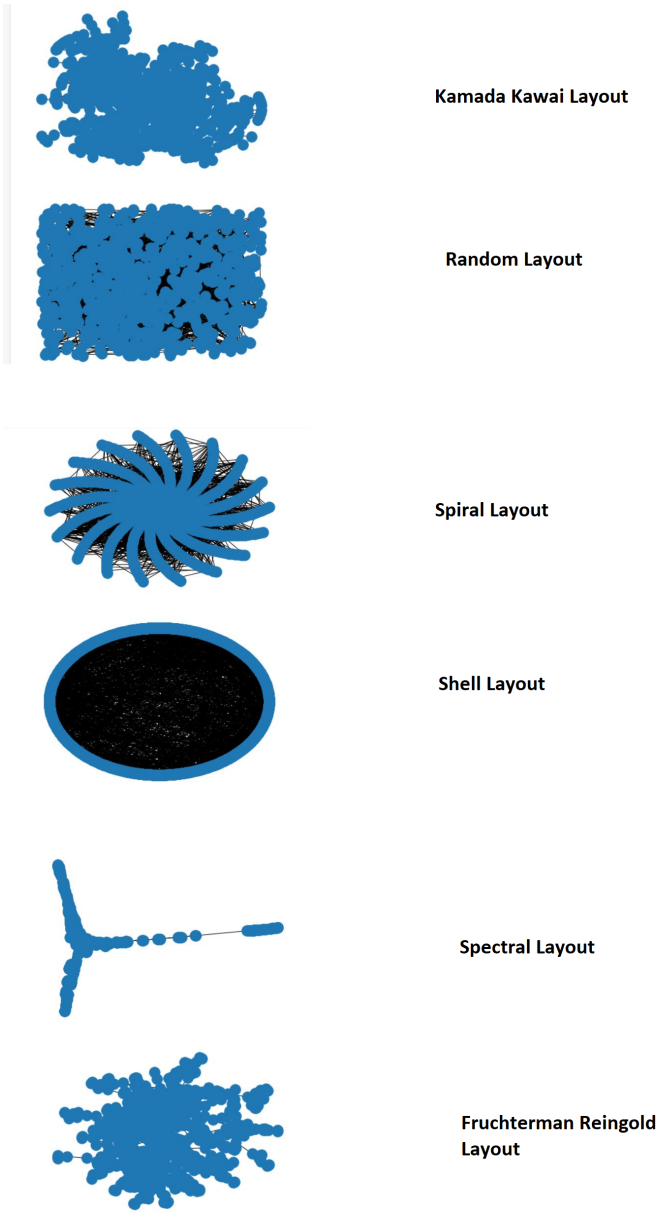
Di seguito, anche per avere una rappresentazione visiva di ciò che verrà analizzato, abbiamo preso in considerazione diversi layout, tutti presi dalla libreria NetworkX:



Spring Layout



Circular Layout



In realtà, nostro malgrado, nessuna di queste rappresentazioni, a meno di particolari layout eccentrici, ha catturato la nostra attenzione, mostrando sempre una configurazione relativamente confusionaria.

2.1 Caratteristiche strutturali

In questa sezione sono state riportate alcune caratteristiche importanti tipiche di una social network, vale a dire:

- Eccentricità: per un nodo n in un grafo G, l'eccentricità di n è il maggiore shortest path[1] possibile tra n e tutti gli altri nodi;
- Diametro: il maggiore shortest path tra una coppia di nodi in un grafo G. È il più grande valore di eccentricità possibile di un nodo;
- Raggio: è il valore minimo di eccentricità di un nodo;
- Periferia: è l'insieme di nodi che hanno la loro eccentricità uguale al loro diametro;
- Centro: è l'insieme di nodi la cui eccentricità è uguale al raggio del grafico.

Inoltre il grafo mostra una densità di 0.005.

Nella seguente Figura 4 si può vedere la sequenza di istruzioni e l'output associato, che vanno a valorizzare quelle suddette misure:

```
NODES:  784
EDGES:  1514
DENSITY:  0.005
CLUSTERING:  0.39010582564457846
Diameter:  21
Radius:  11
Periphery:  ['4263', '4766', '5360', '6905', '6859', '4870', '715', '8282', '8195', '400', '3553', '7071', '5395']
Center:  ['824', '9755', '4846', '1653', '2133', '6610', '7014', '9482', '6700']
```

Figura 4: Misure strutturali

3 Analisi delle centralità

Ad oggi non siamo ancora in grado di definire in modo oggettivo l'importanza di un nodo, anche perché comunque esistono diverse accezioni di "importanza", ognuna delle quali si riferisce ad una particolare caratteristica che un nodo può vantare. Pertanto in questo Capitolo mostriamo l'analisi che abbiamo condotto in relazione alla centralità dei nodi, focalizzandoci sulle diverse possibilità di centralità (ogni centralità viene corredata con 3 layout differenti) e le distribuzioni che ne vengono fuori.

[1] Uno shortest path, o percorso geodetico, tra due nodi in un grafo è un percorso con il numero minimo di archi.

3.1 Degree Centrality

La degree centrality misura la centralità di un nodo in funzione del suo grado, ovvero di quanto esso sia strutturalmente centrale all'interno della rete e così dunque connesso a tanti altri nodi. È generalmente usata per reti molto caotiche. La principale criticità che si può imputare a questa misura è che non tiene conto del "peso" degli archi e si basa solo sul semplice numero di archi, ma riteniamo che in una rete come la nostra dove non c'è un peso associato agli archi, può essere una misura importante al fine dell'analisi. Di seguito viene riportato il grafico di distribuzione della degree centrality, in 3 layout differenti: istogramma, spring layout e spiral layout (Figura 5, 6, 7).

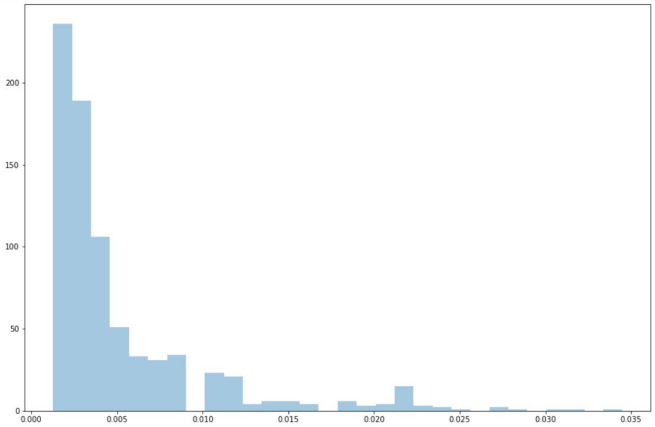


Figura 5: Istogramma della Degree Centrality

Come supportato dalla teoria, possiamo dire con certezza che segue una power law. Pertanto ci sono pochi nodi con una importanza alta e viceversa tanti nodi meno rilevanti, con una importanza sensibilmente inferiore. Ad una conclusione del genere si può giungere anche partendo dalle altre 2 configurazioni, dove al colore giallo si associa una centralità alta (infatti ce ne sono molto pochi) e viola per un valore basso.

3.2 Betweenness Centrality

Una centralità di questo tipo prende in considerazione una concezione diversa di importanza: è basata sull'assunzione che un individuo può guadagnare potere se presidia un collo di bottiglia per la comunicazione (Figura 8, 9, 10).

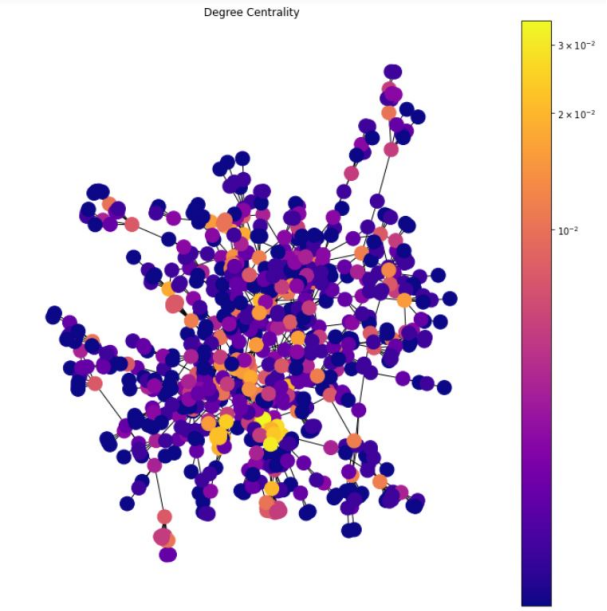


Figura 6: Spring Layout della Degree Centrality

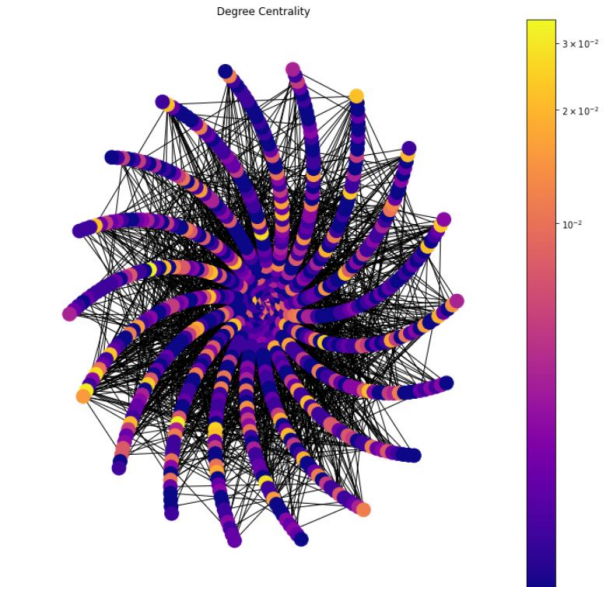


Figura 7: Spiral layout della Degree Centrality

Tutte e 3 le configurazioni traducono la stessa informazione: pochissimi sono i nodi che vantano una betweenness centrality rilevante, a differenza della quasi totalità dei nodi che non ha questa caratteristica.

Un risultato del genere può derivare dal fatto che non ci sono delle

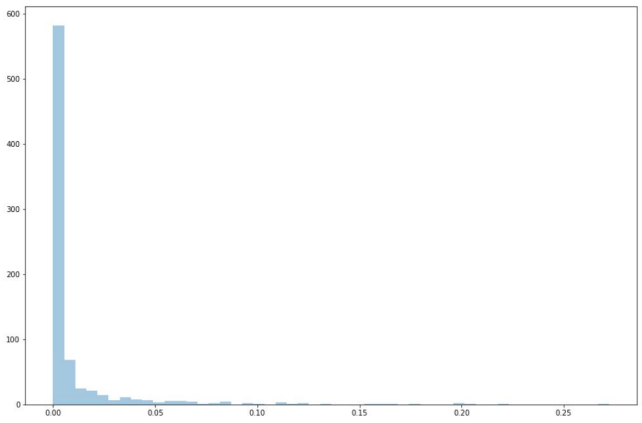


Figura 8: Istogramma della Betweenness Centrality

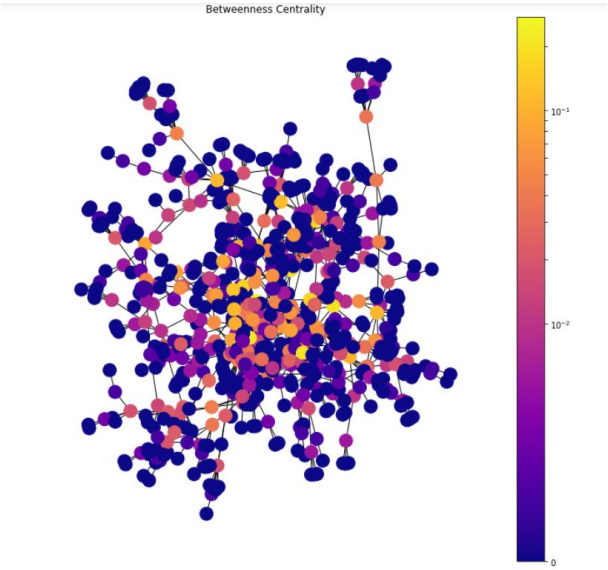


Figura 9: Spring Layout della Betweenness Centrality

comunità che dividono l'insieme dei ricercatori, ma che anzi vantano coesione.

3.3 Closeness Centrality

Come suggerisce il nome, questa metrica per la centralità pone il focus sulla vicinanza di un nodo rispetto agli altri, favorendo quindi quei nodi che hanno virtualmente la capacità di trasportare le informazioni da un lato all'altro della rete.

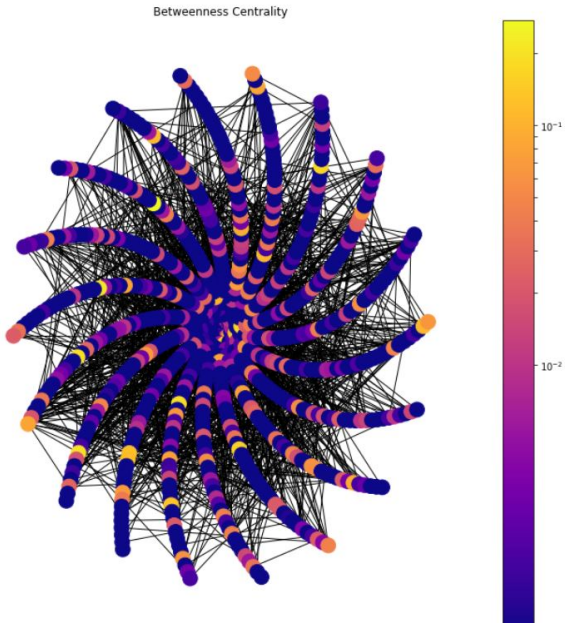


Figura 10: Spiral Layout della Betweenness Centrality

Al fine di garantire la massima chiarezza nell'esposizione, apponiamo la definizione di distanza, la misura su cui fa fulcro questa centralità: la distanza tra due nodi è il numero minimo di archi che bisogna attraversare per raggiungere uno dei nodi partendo dall'altro (Figura 11, 12, 13).

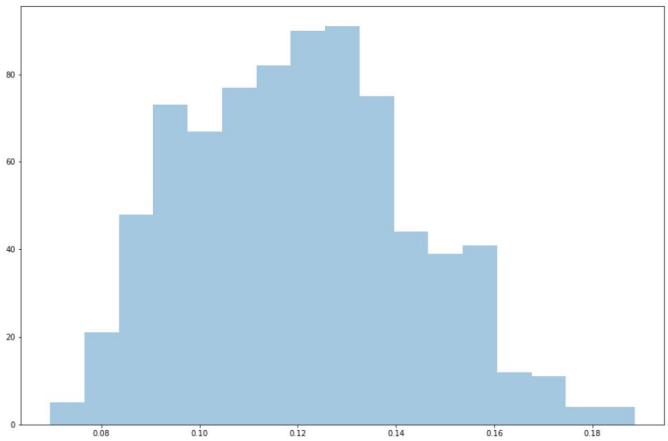


Figura 11: Istogramma della Closeness Centrality

Stavolta la distribuzione non segue una power law, piuttosto una curva assimilabile ad una gaussiana, con quindi valori bassi "ai lati" (ovvero nei

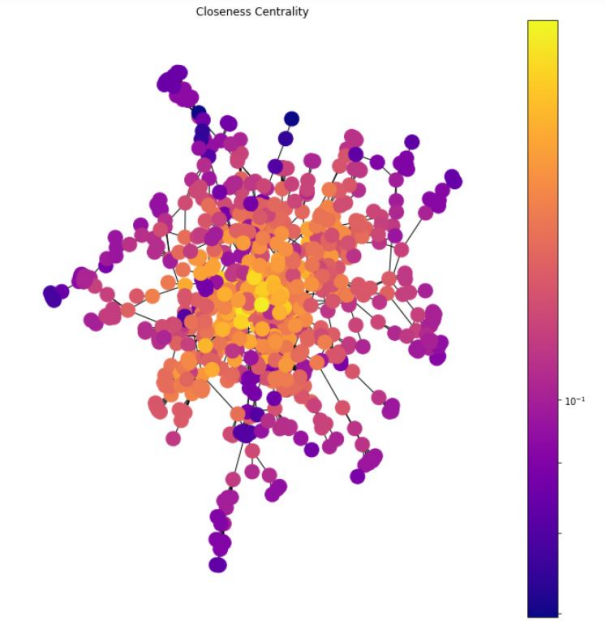


Figura 12: Spring Layout della Closeness Centrality

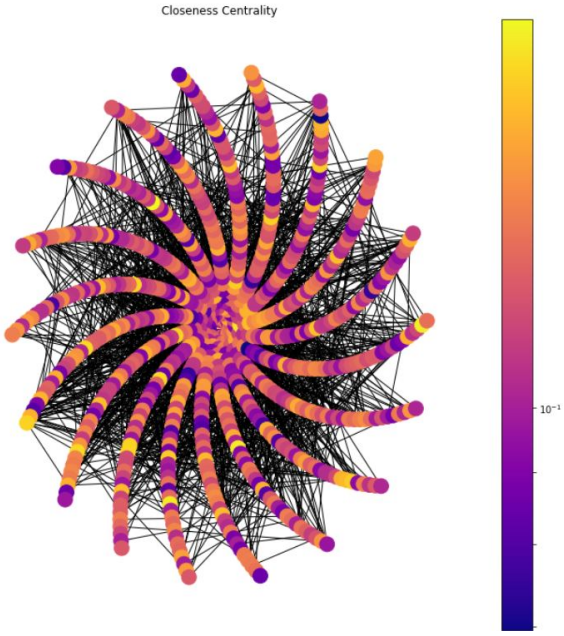


Figura 13: Spiral Layout della Closeness Centrality

due estremi destro e sinistro) e dei valori invece abbastanza rilevanti nella zona centrale. La curva mostrata qui sopra evidenzia come ci sia un buon

numero di nodi con un valore di closeness centrality in un range compreso tra 0.09 e 0.13. Dopotutto, un risultato di questo tipo era prevedibile: del resto in precedenza avevamo introdotto la rete definendola come abbastanza confusionaria, pertanto appare ragionevole che la maggior parte dei nodi sia collegata bene con tutto il resto della rete, e che il numero di nodi "abbastanza isolati" sia relativamente contenuto, anche perché comunque ci siamo soffermati solo sulla componente connessa. Una informazione di questo tipo viene avvalorata anche dagli altri 2 layout giustapposti: entrambi mostrano come, seguendo la colorazione indicata nella legenda, nei grafi prevalga una colorazione tendente al giallo mentre i nodi con una closeness bassa, di colorazione viola, sono rari.

3.4 Eigenvector Centrality

L'ultima misura che abbiamo deciso di prendere in considerazione in merito alla centralità prende il nome di "eigenvector centrality". Un nodo è centrale se è connesso a molti nodi che sono, a loro volta, centrali.

Tale misura può essere spiegata con l'ausilio di un esempio molto chiaro: se avessimo una rete della famiglia Corleone della trilogia de "Il Padrino", di sicuro Don Vito Corleone (Figura 14) avrebbe una posiziona marginale. Dopotutto, egli aveva contatti solo con i suoi fidati più stretti, e quindi, se ci limitassimo a valutarlo secondo le metriche di centralità presentate in precedenza, nessuna di esse evidenzierebbe un valore importante per il suo nodo.

Figura 14: Don Vito Corleone

Ma inutile dire che la figura di Don Vito era assolutamente centrale per la famiglia, e tutte le decisioni di più alto grado passavano attraverso la sua approvazione. Pertanto la metrica dell'eigenvector centrality va alla ricerca di quelle che vengono chiamate "eminenze grigie": quelle figure che sono apparentemente laterali nella rete, ma che sono in realtà molto importanti se consideriamo il fatto che sono legate a figure più centrali .

Dalle figure 15, 16, 17, notiamo (ponendo l'accento sull'istogramma) che sono davvero pochi i nodi con un valore di centralità elevato. Possiamo

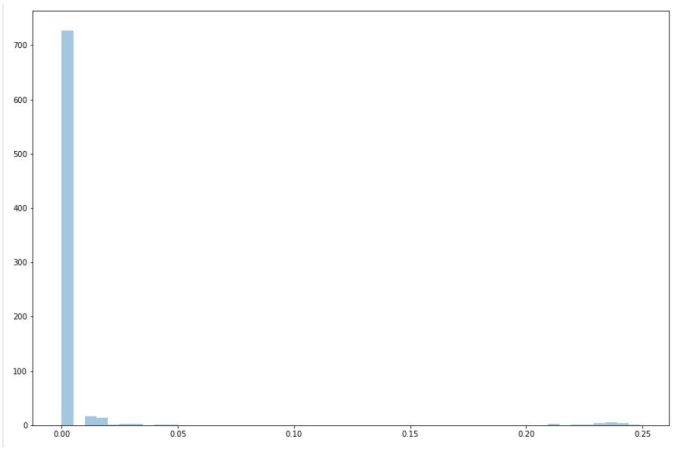


Figura 15: Istogramma della Eigenvector Centrality

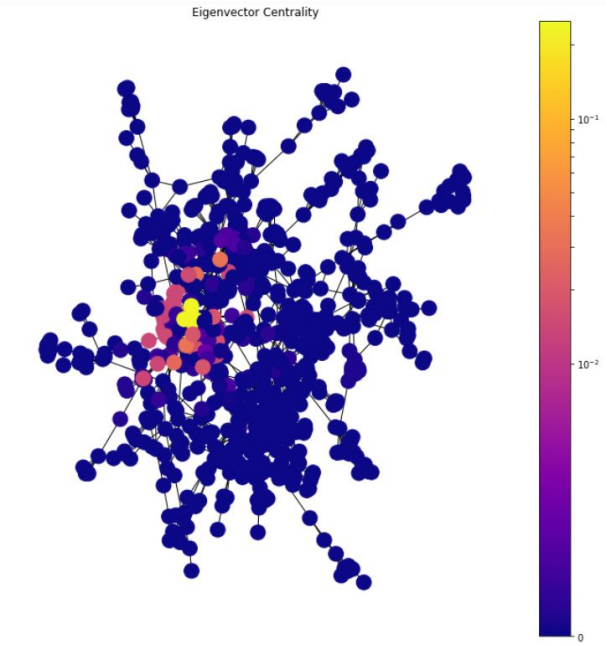


Figura 16: Spring Layout della Eigenvector Centrality

giustificare un risultato di questo tipo ipotizzando che quei nodi con un valore di centralità alto siano i pionieri della ricerca in quest'ambito e che quindi vengano chiamati spesso per delle collaborazioni scientifiche.

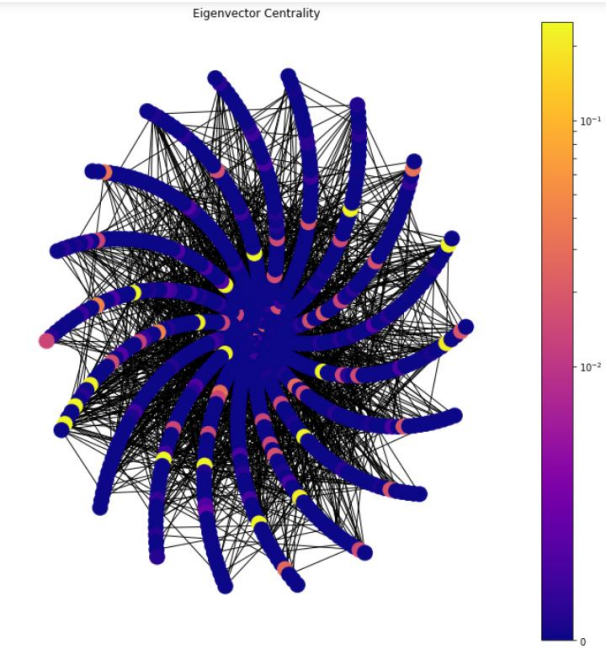


Figura 17: Spiral Layout della Eigenvector Centrality

3.4.1 Ego Network

A questo punto abbiamo proceduto ad ottenere il nodo con il valore massimo della eigenvector centrality (che poi è risultato essere il nodo con indice 2741) e abbiamo poi visualizzato la ego network[2] associata, centrata su quel nodo.

Come si può vedere dalla Figura 18, il nodo con indice 2741 ricopre una posizione centrale nella ego network, e, come deve essere per costruzione, tutti gli altri nodi della rete fanno riferimento a lui.

Infine, per verificare la bontà dei risultati ottenuti, di seguito (Figura 19, 20) mostriamo il grafico della closeness centrality della ego network: il nostro obiettivo era appunto mostrare che la closeness centrality era pari a 1 per il nodo centrale della ego, proprio per definizione della centrality presa in considerazione, e così è.

[2]Le reti dell'ego sono costituite da un nodo centrale ("ego") e dai nodi a cui l'ego è direttamente connesso (questi sono chiamati "alter").

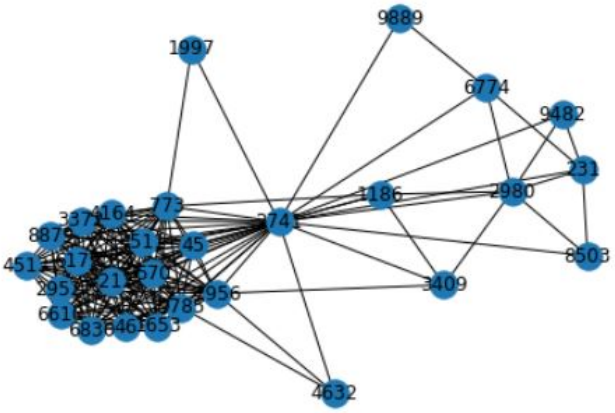


Figura 18: Ego Network

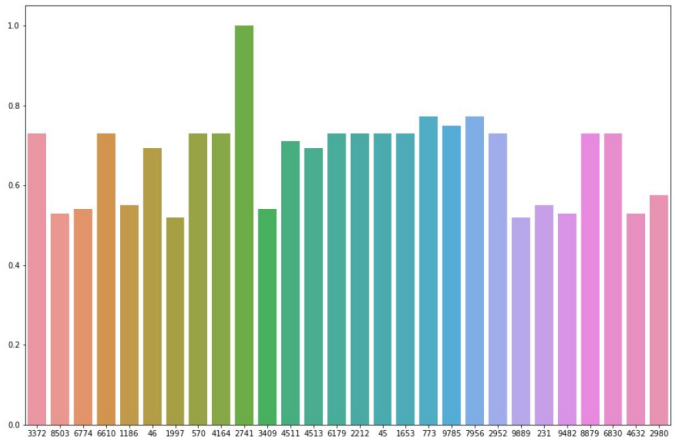


Figura 19: Closeness Centrality della Ego Network

4 Analisi triadi e clique

4.1 Triadi

Una triade non è altro che un insieme di tre nodi, dove la struttura più stabile nel tempo è la triade chiusa, ovvero una triade in cui ogni nodo è connesso con gli altri.

Nello specifico abbiamo utilizzato la funzione triadic census di NetworkX, che dà in output il censimento dei 16 possibili tipi di triadi presenti nella rete. Le 16 triadi possibili sono le seguenti (Figura 21):

Chiaramente nel nostro caso, avendo un grafo indiretto (gli archi non

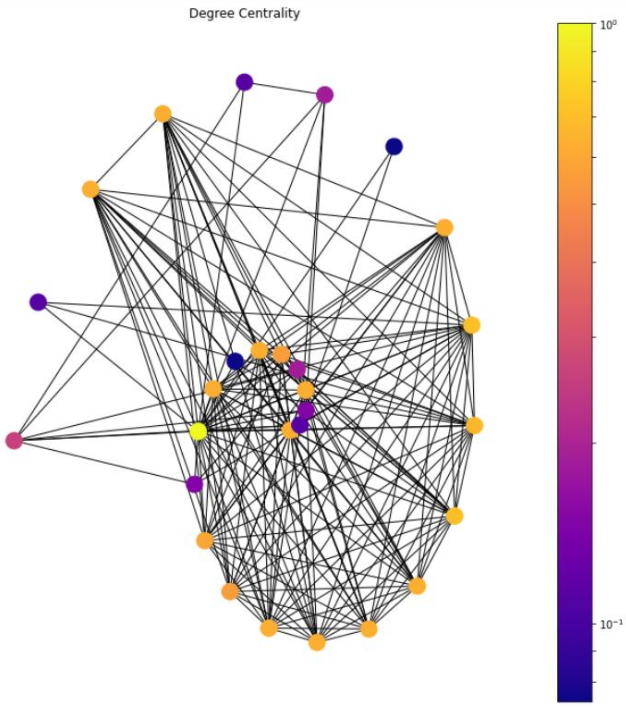


Figura 20: Spiral Layout della Degree Centrality della Ego Network

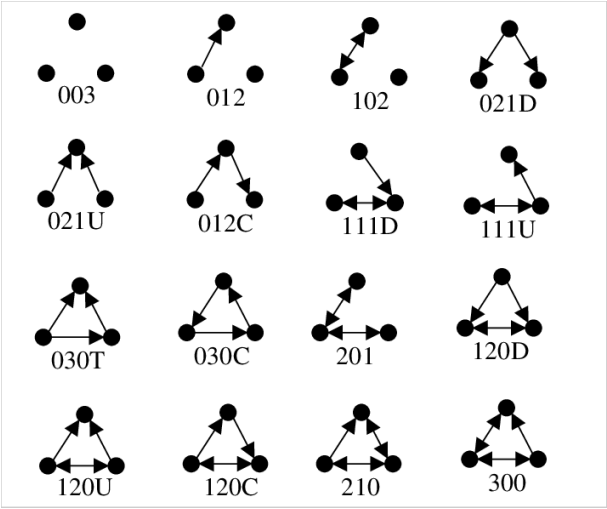


Figura 21: Possibili triadi in un grafo diretto

hanno un verso) il numero di triadi si riduce sensibilmente (ed infatti in seguito vedremo come questa ipotesi viene confermata dal codice stesso). Infatti non avendo un verso, le triadi che possiamo trovare sono solo quelle

indicate dai codici “003”, “012”, “201” e “300”.

Pertanto il censimento conta solo quei 4 tipi di triadi, dando in output ciò che si può vedere nella Figura seguente (Figura 22):

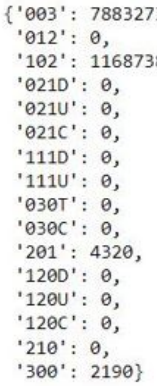


Figura 22: Censimento delle triadi

Come anticipato, sono solo 4 le triadi contate, ed in particolare notiamo come la maggior parte siano triadi del tipo “003”, ovvero una terna di nodi non direttamente connessi. Invece saranno degne di nota quelle 2190 triadi del tipo 300, ovvero le triadi chiuse (completamente connesse) che ci interessano perché le più stabili nel tempo.

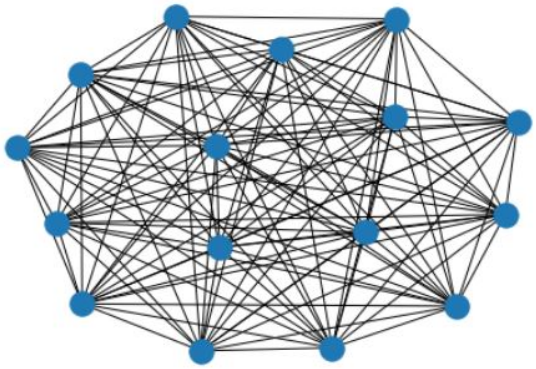
4.2 Clique

Allargando l’orizzonte, abbiamo deciso di prendere in considerazione le clique. Per clique, nella teoria dei grafi, si intende un insieme di vertici in un grafo non orientato, tale che, per ogni coppia di vertici, esiste un arco che li collega.

Di nuovo, NetworkX ci viene in aiuto anche in quest’ambito: in particolar modo ci serviamo di due metodi messi a disposizione dalla libreria, ovvero enumerate all clique e find cliques. Il primo metodo restituisce semplicemente tutte le clique del grafo passatogli come parametro, e nel nostro caso ha individuato la bellezza di 182824 elementi.

Il secondo svolge un compito leggermente piu specifico: restituisce tutte le clique massimali[3], e qui ce ne sono ben 620.

Di seguito, data l’importanza legata al nodo 2741, abbiamo deciso di verificare che facesse parte di una clique che effettivamente poi si é dimostrata



massimale, e di seguito si mostra la forma della struttura (Figura 4.2).

5 Community detection

In questa sezione é stata affrontata la problematica di community detection, vale a dire il cercare di catalogare i nodi in set di nodi chiamati comunitá.

Per appartenere ad una particolare comunitá, il nodo deve condividere con gli altri nodi delle caratteristiche chiave, nel nostro caso il collegamento con gli altri nodi della stessa comunitá. Chiaramente la connessione tra nodi non é l'unica feature che va presa in considerazione, ma nel nostro caso altre possibili feature (genere, localitá, ecc) non sono associabili alla nostra rete, e quindi si é costretti a limitare l'analisi su questa caratteristica. Per tale ragione l'algoritmo tiene in considerazione solo i link tra i vari utenti, definendo una comunitá come un insieme di utenti reciprocamente connessi.

Ciononostante, l'algoritmo greedy di community detection ha dato un risultato che noi riteniamo comunque molto interessante e degno di nota, come mostriamo nella seguente figura (Figura 23):

Se non ci si lascia scoraggiare dalla moltitudine di nodi e colori (inevitabile data la complessitá della rete), in realtà la figura é facilmente interpretabile: ad ogni comunitá abbiamo associato un colore.

Chiaramente le comunitá originariamente identificate dall'algoritmo erano davvero troppe, e non sarebbe stato stimolante rappresentarle. A tal fine, abbiamo preferito concentrarci solo sulle prime 9 comunitá (quelle piú nu-

[3]per ogni nodo n, una clique massima per n indica il sottografo completo maggiore contenente n

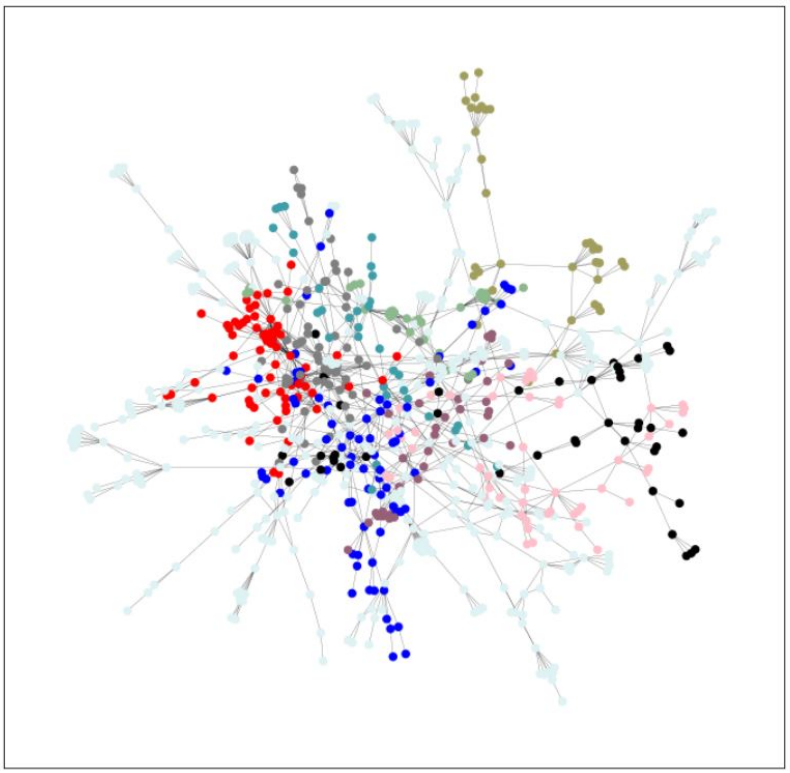


Figura 23: Comunità secondo l'algoritmo greedy

merose) e condensando tutto il resto in un'unica comunitá (evidenziata dal colore bianco ghiaccio).

Dunque dal punto di vista meramente grafico, si puó notare come l'algoritmo effettivamente si sia comportato bene, mostrando delle comunitá abbastanza distinte.

Il risultato chiaramente non puó essere eccellente (si guardi il blu ad esempio, che si espande per tutta la rete), ma ció é facilmente giustificabile, per il fatto che la rete é molto complessa.

Infine concludiamo il nostro lavoro sulla Social Network Analysis facendo uso di un altro algoritmo per la Community Detection, che prende il nome di algoritmo di Girvan-Newman.

Tale algoritmo, la cui relativa funzione si trova all'interno della libreria NetworkX di Python, si basa sull'eliminazione iterativa degli archi che hanno il maggior numero di shortest path tra i nodi che li attraversano. Rimuovendo gli archi dal grafico uno per uno, la rete si scompone in parti piú piccole, che saranno quindi le comunitá.

Applicandolo al nostro caso, esso ha trovato due comunitá, che in figura

mostriamo con colori verde e blu (Figura 24):

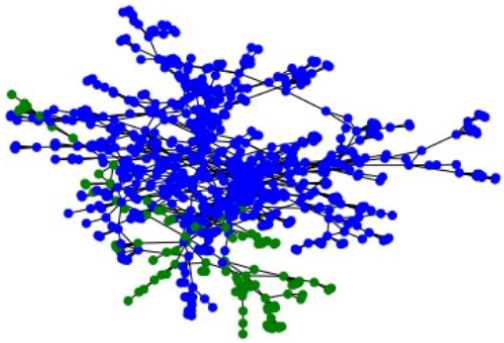


Figura 24: Comunità secondo l'algoritmo di Girvan-Newman

Purtroppo anche graficamente si puó notare come i risultati non siano tra i migliori: ha trovato bene la comunitá verde, ma 2 sono decisamente poche (si guardi la preminenza del blu sulla rete), e quindi possiamo concludere dicendo che strutturalmente la nostra rete non si presta bene all'applicazione di questo algoritmo.