Relazione Primo Progetto Social Computing

Deano Luca - 159357 - 159357@spes.uniud.it
Mancardi Devin - 159108 - 159108@spes.uniud.it
Mauro Gianfranco - 157548 - 157548@spes.uniud.it
Volpi Davide - 157048 - 157048@spes.uniud.it

1 Executive Summary

Dato il dataframe Nodes.csv, riportante i nomi di sette autori e le loro affiliazioni: scaricare tutte le informazioni riguardanti gli autori: ID, citato da e i loro interessi; trovare tutti i relativi coautori degli autori originali e di essi scaricare le stesse informazioni come fatto precedentemente.; creare un dataframe di due colonne contenente le relazioni di coautorship fra gli autori originali e i coautori, rinominandolo Edges.csv.
Per aggiornare e modificare i dataframe abbiamo usufruito della funzionalità della libreria Pandas, mentre per scaricare i dati relativi ai vari autori e coautori, la libreria SerpApi. Il fine di queste operazioni è visualizzare visivamente le relazioni di coauthorship attraverso la creazione di un grafo sfruttando le funzionalità della libreria NetworkX. Il passo successivo è stato aggiornare il grafo tramite il metodo del preferential attachment, aggiungendo 50 archi per vedere come sarebbe cambiato. Per evidenziare al meglio le differenze strutturali tra i due grafi abbiamo calcolato varie misure, sia in generale che sui singoli nodi:

- Coefficiente di clustering medio, Raggio, Distanza Media. . . ..per i grafi generali
- Degree Centrality, Betweennees Centrality, PageRank. . . ..per i nodi

L'ultimo punto consisteva nel produrre una visualizzazione interattiva via html dei due grafi utilizzando la libreria PyVis. Da come si può evincere dai risultati ottenuti dalle analisi, si denota una differenza tra i due grafi, quello normale e quello andatosi a creare con la tecnica del Preferential attachment.

2 Metodologia

2.1 Scaricamento Dei Dati

Utilizziamo la libreria SerpAPI per effettuare ricerche sui profili Google Scholar degli autori specificati e ottenere informazioni come *author_id*, *cited_by*, e *interests*.
Usiamo SerpAPI come servizio per ottenere i dati da Google Scholar che semplifica l'estrazione di informazioni da risultati di ricerca su Google e da altri motori di ricerca. Con una semplice implementazione di un ciclo for abbiamo fatto in modo di eseguire la ricerca per ogni riga di nodes.csv (ovvero di ogni autore). Per eseguire una corretta ricerca dei profili Google Scholar sono stati usati i campi *author_id* e *affiliations* per costruire la query, andando a cercare il profilo corretto dell'autore. Dopo aver ottenuto i risultati dalla richiesta SerpAPI, abbiamo iterato sui profili degli autori ("profiles") per estrarre le informazioni di interesse come *author_id*, *cited_by* e *interests*.
Per gestire la lista di interessi multipli la nostra scelta è stata quella di inizializzare una lista vuota (*interest_list*) dei titoli di interesse per poi convertirla in una stringa separata da virgole. In seguito viene aggiornato il dataframe ad ogni iterazione con il metodo ATper salvare i file su un CSV.

2.2 Espansione Del Grafo

Per eseguire le analisi aggiuntive abbiamo applicato la tecnica del Preferential Attachment sul grafo esistente per generarne uno nuovo con un aumento degli archi di 50 in base ai punteggi ottenuti.
Il principio menzionato indica che i nodi con un alto grado sono più inclini a ricevere nuove connessioni rispetto ai nodi con un grado più basso.

1. Viene creato un nuovo grafo *G_PREFERENTIAL_ATTACHMENT* inizializzato con gli stessi nodi del grafo esistente
2. Si calcolano i punteggi di Preferential Attachment per tutte le coppie di nodi del grafo usando la funzione *"Nx.Preferential_attachment"*, ordinati in modo decrescente.

3. Per aggiungere i nuovi archi al grafo *G_Preferential_Attachment* viene specificato un numero (*num_edges_to_add*) e le prime coppie con punteggi più alti vengono aggiunti.

4. Il nuovo grafo viene salvato nel file *'graphs/graphs_preferential_attachment.pickle'*.

5. Viene eseguita la visualizzazione del nuovo grafo con etichette e colori diversi per i nodi con diversi gradi.Nodi con un grado più elevato hanno una colorazione diversa rispetto a quelli con un grado più basso.

L'obiettivo principale è simulare il processo di crescita dei grafi secondo il principio del Preferential Attachment, spesso osservato in reti reali come le reti sociali o le reti di citazioni accademiche.
Si noti che nel codice è possibile trovare due metodi diversi di Preferential Attachment, il primo è stato implementato prendendo i 50 archi più probabili non tenendo conto che ogni qual volta un arco viene aggiunto al grafo si vada a modificare anche il grado dei due nodi collegati ad esso; nel secondo caso quest'ultimo è stato implementato tenendo conto di questa casistica e in più facendo una normalizzazione tenendo conto del grado del nodo, così facendo nodi con grado più basso posso essere considerati candidati a qui aggiungere un arco.

2.3 Costruzione e Visualizzazione dei Grafi

Per la costruzione dei grafi non diretti abbiamo usato la funzione *"from_pandas_edgelist"* di NetworkX, che prende gli archi del dataframe "edges.CSV" specificando le colonne "author" e "coauthor" come sorgente e destinazione degli archi.
Dopo la creazione del grafo vengono assegnati dei colori diversi ai nodi in base al loro grado per crearne uno interattivo che evidenzia le caratteristiche chiave della rete.

- I nodi sono colorati in base al numero di archi incidenti su di essi (il grado): grado 1 grigio, tra 2 e 10 blu, tra 11 e 20 viola, superiore a 20 giallo
- Il posizionamento dei nodi è stato calcolato utilizzando l'algoritmo di disposizione *"spring_layout"* di NetworkX. Questo algoritmo posiziona i nodi in modo tale da minimizzare la forza di attrazione e repulsione tra di essi, offendo una visualizzazione chiara della struttura di rete.
- I nodi del grafo sono etichettati con i loro identificatori univoci (*name*)
- La dimensione è stata fissata a 100 per nodi e 1 per gli archi, così da garantire una visualizzazione chiara senza sovrapposizioni.

2.4 Ulteriori Assunzioni

Per trovare e scaricare la lista dei coautori di ogni autore, abbiamo effettuato un' ulteriore ricerca utilizzando direttamente i loro *author_id*. Per ogni ricerca abbiamo inserito in un nuovo dataframe il nome del coautore, il suo *author_id* e le *affliliations*. La decisione di salvare fin da subito queste tre informazioni non è superflua. Il passo successivo infatti, richiede di salvare nel dataframe tutte le informazioni che abbiamo salvato anche per i 7 autori originali; avendo già a disposizione nome e affiliazione possiamo eseguire una ricerca "corretta" dei coautori ed evitare in questo modo di selezionare il primo della lista (che potrebbe risultare errato).

3 Risultati

3.1 Dataframe nodes.csv

Il dataframe finale è un update di quello fornitoci ad inizio progetto con l'aggiunta dei risultati ottenuti dalle ricerche SerpAPI (le nuove colonne sono: *Author_ID*, *Cited_by* e *interests*).

3.2 Dataframe edges.csv

Il primo dei due risultati ottenuti durante questa fase è la concatenazione del dataframe nodes.csv con uno contenente le informazioni relative ai coautori dei 7 autori originali (le informazioni scaricate sono le stesse del punto 3.1).
Il secondo risultato fa riferimento alla creazione di un dataframe chiamato edges.csv e che contiene la relazione di coautorship.

3.3

Questo punto richiedeva di rappresentare in modo grafico e visuale le relazioni di co-autorità tra gli autori della rete accademica considerata. La visualizzazione evidenzia i nodi più centrali della rete attraverso la colorazione, facilitando l'identificazione di autori e coautori con un ruolo significativo nella collaborazione scientifica. La disposizione dei nodi offre insights sulla struttura della rete, con nodi più centrali posizionati in modo strategico per massimizzare la connettività.
Questa visualizzazione può essere utilizzata per analizzare la topologia della rete degli autori e facilitare l'identificazione di nodi chiave o gruppi di collaborazione all'interno della rete sociale rappresentata.

3.4

L'espansione del grafo attraverso la tecnica del Preferential Attachment fornisce un'opportunità per esplorare nuove connessioni potenziali nella rete di co-autorità. La visualizzazione del nuovo grafo consente di identificare eventuali nodi centrali e di comprendere meglio la struttura complessiva della rete.

3.5 Misure nei due grafi

	Nome Grafo	Clustering Medio	Centro del Grafo
0	Grafo normale	0.165125	Kevin Roitero, Stefano Mizzaro...
1	Grafo Preferential Attachment	0.208064	David La Barbera, Kevin Roitero...
2	Grafo Preferential Attachment corretto	0.413358	David La Barbera, Kevin Roitero...

Raggio	Distanza Media	Transitività	Omega	Sigma
2	2.651743	0.147939	0.004855	0.929919
2	2.403623	0.292600	0.002930	0.999118
2	2.427546	0.185083	0.136521	1.015548

Coefficiente di Clustering Medio: misura quanto i vicini di un nodo sono collegati tra loro. Un valore più alto indica una maggiore coesione nella rete. Da come possiamo notare e da come ci aspettavamo, il grafo con 50 archi in più ha una maggiore coesione rispetto al grafo originale.

Centro del Grafo: indica il nodo più vicino a tutti gli altri nodi della rete in termini di lunghezza dei cammini minimi. Questa misura è particolarmente significativa per la valutazione della struttura globale del grafo. Notiamo che il grafo con 50 archi in più ha più centri rispetto a quello originale essendo maggiormente connesso.

Raggio: rappresenta la distanza massima tra il nodo centrale (calcolato come il centro del grafo) e qualsiasi altro nodo nella rete; riflette la dimensione massima della rete. Notiamo che i due grafi hanno dimensione uguale.

Distanza Media: è la lunghezza media dei cammini più corti tra tutti i nodi; indica la tipica distanza tra i nodi nella rete. É una metrica importante per valutare l'accessibilità e la connettività della rete. Dai nostri due valori, possiamo osservare come il primo grafo presenti una distanza media maggiore e alcuni cammini minimi più estesi, mentre il secondo abbia una connettività superiore tra i suoi nodi, con una distanza media inferiore rispetto a quella del grafo originale.

Transitività: misura la probabilità che i vicini di un nodo siano a loro volta collegati tra loro. Valori più alti indicano una maggiore coesione. Notiamo che il grafo con 50 archi in più ha una maggiore coesione.

Coefficiente Omega (ω) e Sigma (σ): i coefficienti Omega e Sigma sono utilizzati per stimare la "small-world-ness" di una rete. Una rete small-world è caratterizzata da una connettività veloce tra i nodi (cammini medi brevi) e, allo stesso tempo, da una struttura fortemente clusterizzata.

- **Valori di Omega più vicini a 0** valori prossimi a 0 indicano che G ha caratteristiche della small world ness. Valori negativi indicano che G è simile a un reticolo mentre valori positivi indicano che G è un grafo casuale. indicano Per tutti e tre i grafi troviamo un valore di Omega simile (vicino allo 0).
- **Valori di Sigma maggiori a 1** indicano che il G è comunemente classificato come small-world. Come per Omega, i tre valori sono molto simili (vicini ad 1).

Nel contesto dei dati forniti, il secondo grafo (Grafo Preferential Attachment) avendo un valore maggiore di 1 di Sigma,ovvero un coefficiente di clusetring relativamente alto, e un omega tendente a zero, ovvero una lunghezza media dei percorsi breve, sembra essere il più vicino al concetto di "small-world".

3.6

Per calcolare le centralità per ogni nodo in entrambi i grafi, utilizzeremo diverse metriche di centralità, tra cui:

Degree Centrality: indica quanti archi sono collegati a un nodo. Un nodo con molti archi ha una degree centrality più alta. Da come possiamo notare, i 7 autori originali (in particolare i nodi gialli) avranno un valore più elevato rispetto agli altri.

Betweenness Centrality: misura il numero di volte in cui un nodo si trova nei cammini minimi tra altri nodi. I nodi con alta betweenness centrality svolgono un ruolo critico nel collegamento di diverse parti della rete. Come per la Degree Centrality i nodi gialli in entrambi i grafi svolgono un ruolo critico.

Closeness Centrality: indica la vicinanza di un nodo rispetto a tutti gli altri nodi nella rete. Un nodo con alta closeness centrality è più vicino a tutti gli altri nodi. I nodi con più collegamenti che avranno riscontrato una alta degree e betwnees centrality sarannno gli stessi che avranno valori elevati in questa metrica

PageRank: algoritmo di ordinamento dei motori di ricerca che assegna un punteggio numerico a ciascun elemento di un insieme di documenti ipertestuali, con l'obiettivo di misurare l'importanza relativa dei documenti.

HITS (Hyperlink-Induced Topic Search): divide la centralità in due metriche:

- Hubness: quanto il nodo è collegato ad altri nodi. Un nodo con un alto punteggio di hubness è considerato un hub nella rete.
- Authority: quanti collegamenti riceve il nodo da parte di altri nodi. Un nodo con un alto punteggio di authority è considerato un'autorità nella rete.

HITS fa si che i nodi possano essere distinti tra quelli che sono più adatti nel collegare (hub) e quelli maggiormente indicati per ricevere collegamenti (authority).

3.7

Il risultato di questa sezione è stato la creazione del grafo interattivo del web. Si può notare che passando con il mouse sopra ad un nodo, si possono visualizzare i molteplici attributi di quest'ultimo. Cliccando poi i nodi con il mouse e spostandolo, si può cambiare la posizione di ognuno di essi a piacere (entro una certa fisica che abbiamo impostato noi tramite codice).

4 Conclusioni

In sintesi, dato un dataframe contenente sette autori scaricare le informazioni richieste; successivamente, effettuare un'ulteriore ricerca per identificare e acquisire dati relativi ai loro coautori. Aggiornare il dataframe nodes.csv e crearne un'altro (edges.csv), per poi creare due grafi. Inoltre, sono stati calcolati diversi parametri di rilevanza sia per i due grafi nel loro complesso che per i singoli nodi. Realizzare un grafo interattivo visualizzabile su browser.
Ogni passaggio richiesto è stato svolto in maniera corretta, non ci sono stati errori ne intoppi e dai dati ottenuti ci possiamo considerare soddisfatti. Come possiamo vedere dai punti 3.5 e 3.6 della relazione le misure rispettano a pieno le aspettative che ci eravamo prefissati alla creazione dei due grafi.
Anche i grafi stessi, grazie all'inserimento nel codice di alcune funzioni per la gestione della posizione dei nodi, evidenziano in maniera efficace la differenza tra autori, coautori comuni tra essi e coautori "marginali".
Nel secondo metodo che abbiamo implementato per ricreare il Preferential Attachment corretto, si potrebbe andare a creare un'inconsistenza con i risultati delle misure di analisi di quest'ultimo, perchè ogni qualvolta si esegue il codice di creazione del grafo, essendo implementato con la libreria random per l'aggiunta degli archi, i valori variano ad ogni iterazione.
La decisione di implementare questo metodo è stata fatta nel momento in cui abbiamo notato che il metodo visto in classe del Preferential Attachment non considera l'aggiornamento dei gradi dei nodi e non normalizza la probabilità in base al grado.
Alla fine del progetto ci è sorto un dubbio riguardante il concetto di small world. A prima vista sembra paradossale avere in contemporanea un coefficiente di clustering elevato (i nodi nel grafo sono in qualche modo disconnessi e formano diversi hub da soli) e una lunghezza del cammino minimo medio breve; un hub dovrebbe percorrere una lunga strada per raggiungere un altro hub distante. Così abbiamo voluto approfondire la questione e cercato online delle risposte.
Il motivo per cui entrambe le condizioni possono essere soddisfatte è che nella nostra rete sociale, anche se tutti hanno i loro co-autori più stretti, ci sono diverse persone nella rete che sono estremamente socievoli e servono da collegamento tra diversi hub; l'esistenza di queste persone sociali spiega il piccolo percorso medio breve. In altre parole, due co-autori che non interagiscono fra di loro, molto probabilmente conoscono certe persone "sociali".