Davide Guidobene 877150 01/02/2022

Report Progetto Social Network Analysis

In questo progetto andremo a esaminare i dati rilasciati da Netfilx per la competizione The NetflixPrize del 2006 e in particolare a indagare la presenza eventuali similarità tra film che hanno ricevuto recensioni simili.

Linguaggio usato

Il progetto è stato svolto quasi interamente nel linguaggio R su RStudio per rispecchiare l'approccio presentato a lezione e il file principale del progetto è *SNAProjectNetflixPrize.Rmd*.
L'unica parte del progetto svolta in Python (su Jupyter Notebook) è quella che ha richiesto l'uso di API per richiedere dati al sito Imdb. Questa scelta rispecchia sempre l'approccio presentato a lezione dove ci è stato fortemente consigliato l'utilizzo di Python per le API, inoltre la libreria imdb usata non era disponibile in R. Quest'ultima parte è comunque molto breve e tecnica e non tocca direttamente l'analisi dei dati ed è descritta interamente al paragrafo Data Preprocessing 2.

Data Preprocessing

Il dataset in *combined_data.txt* consiste in una lista di recensioni, dove ogni recensione è concettualmente una quadrupla composta da:

- movie_id: identificativo unico dei film
- user_id: identificativo unico degli utenti
- rating: recensione del film rilasciata dall'utente, può andare da 1 a 5 stelle
- date: data della recensione

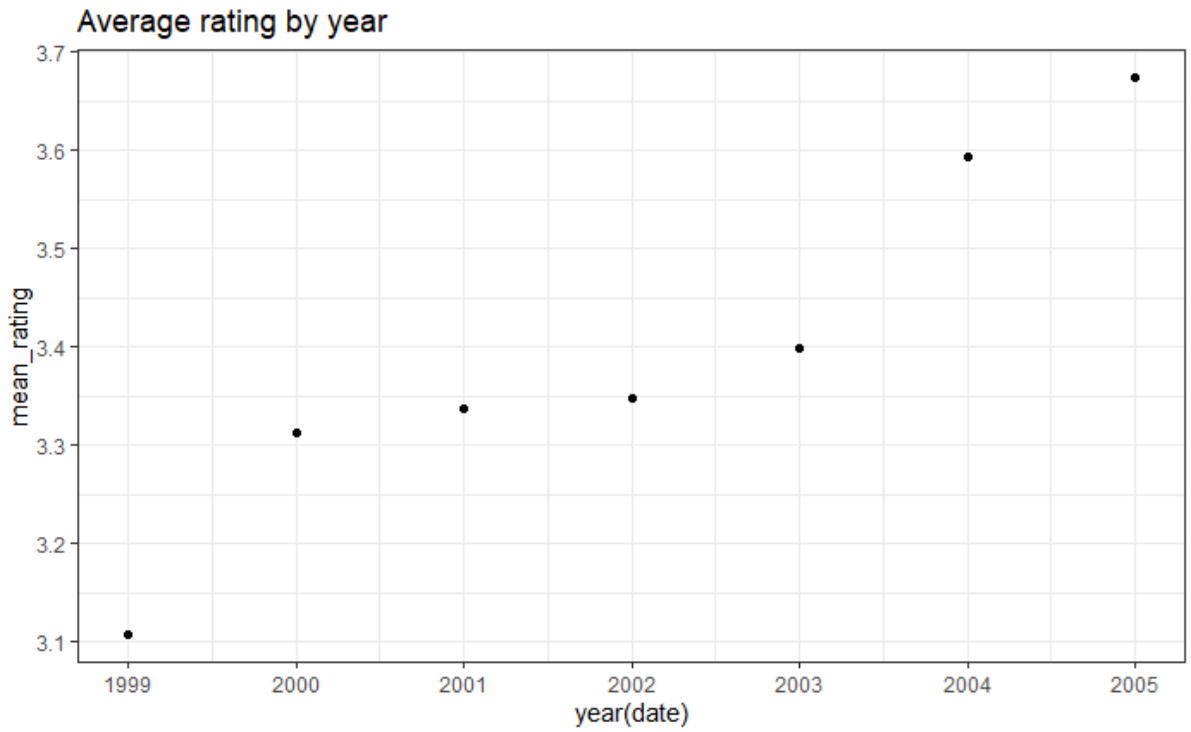
Il file grezzo rilasciato da Netflix presenta le recensioni nella forma *'movie_id'*: seguito da una lista di triple *'user_id' 'rating' 'date'* riferite al 'movie_id'.

V1 <chr>	V2 <int>	V3 <chr>
1:	*NA*	
1488844	3	2005-09-06
822109	5	2005-05-13
885013	4	2005-10-19
30878	4	2005-12-26
823519	3	2004-05-03

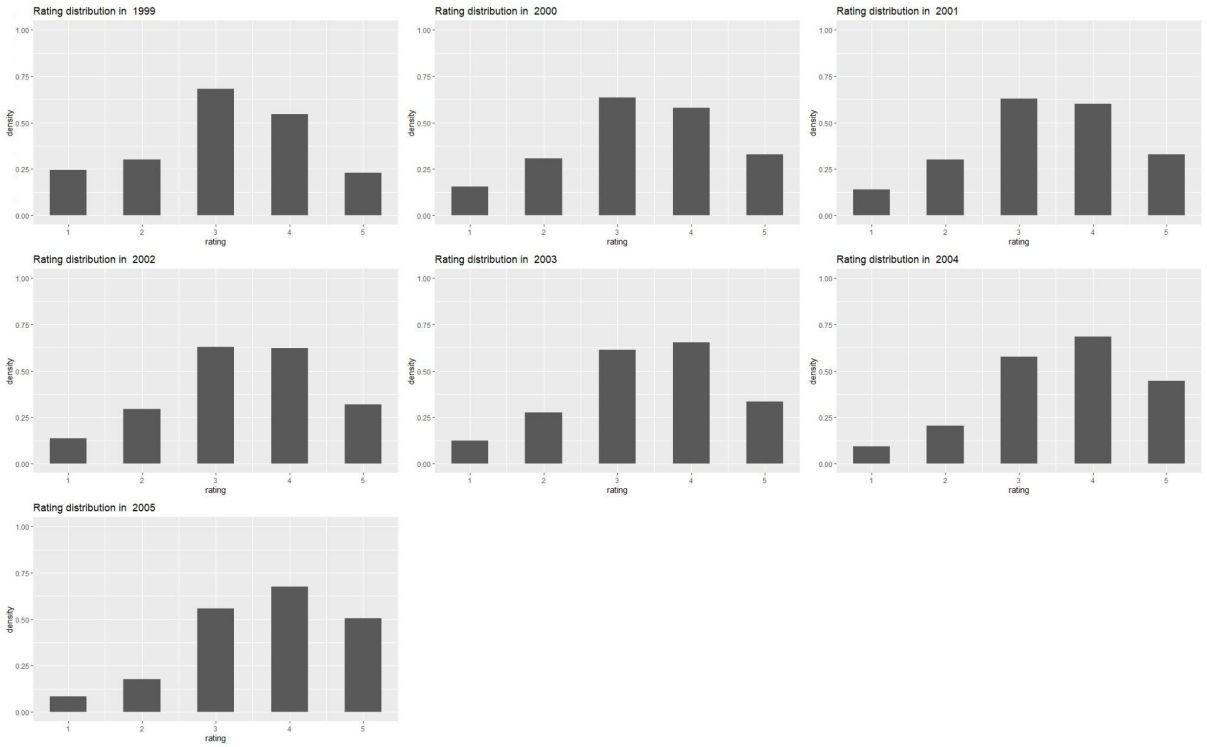
Quindi, la prima cosa da fare è trasformare questa quadrupla concettuale in una quadrupla vera e propria e sistemare il formato dei dati (in modo che id e rating siano dati di tipo numeric e date di tipo Date).

	movie_id <dbl>	user_id <dbl>	rating <dbl>	date <date>
1	1	1488844	3	2005-09-06
2	1	822109	5	2005-05-13
3	1	885013	4	2005-10-19
4	1	30878	4	2005-12-26
5	1	823519	3	2004-05-03
6	1	893988	3	2005-11-17

Siccome il nostro punto di partenza sarà quello di creare una rete bipartita user-movies basandoci sulle recensioni, è importante assicurarci che non ci siano anomalie sul valore delle recensioni.
Quindi una prima cosa da fare per visualizzare i dati è plottare l'andamento delle recensioni nel tempo.
La prima figura è un semplice grafo a dispersione.



Nella figura seguente possiamo osservare lo stesso fenomeno nella forma di istogrammi di densità, adatti a questo genere di rappresentazione in quanto le recensioni sono variabili categoriali.
Per ogni rating (da 1 a 5) la corrispondente colonna dell'istogramma rappresenta il rapporto di recensioni con quel rating sul numero totale di recensioni di quell'anno.



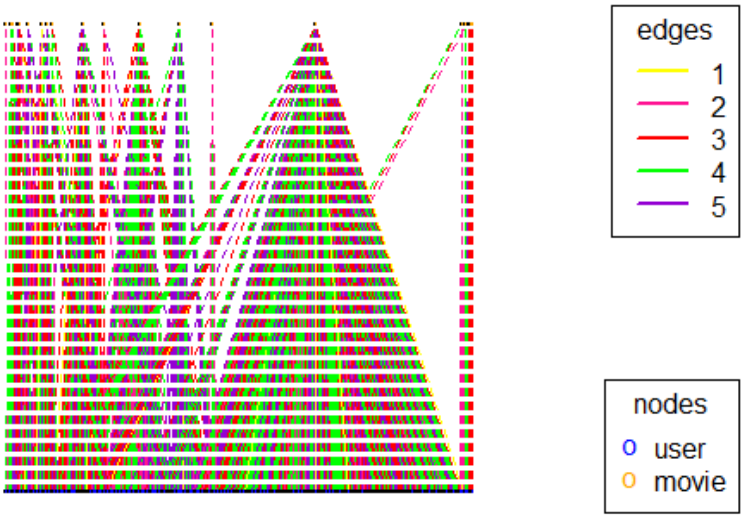
Notiamo che nel 2004 le recensioni hanno subito un incremento improvviso, nel grafico a dispersione c'è un incremento evidente mentre negli istogrammi notiamo che nel 2004 e nel 2005 le colonne che rappresentano i rating 4 e 5 sono significamente cresciute, mentre le altre 3 sono decrementate. Cos'è successo?
Dopo una breve ricerca internet ho infatti scoperto che nel 2004, Netflix ha effettuato un cambiamento nel suo rating system: il testo che accompagna ciascuna possibile recensione è passato da una scala più oggettiva (eccezionale, buono, …) a una più soggettiva (mi è piaciuto, non mi è piaciuto, …).
Questo cambiamento sembra aver incoraggiato gli utenti a dare recensioni più alte anche a titoli meno importanti ma che sono piaciuti.
In seguito a queste analisi ho ritenuto opportuno tenere solamente le recensioni successive al 2004 ed eliminare le altre. Lo svantaggio principale di questa scelta è la perdita di alcuni dati, ma la maggior parte delle recensioni sono comunque del 2005, inoltre, anche con meno dati, molte delle mie analisi future dovranno comunque lavorare su un campione della popolazione per questioni di efficienza dato che la rete che si andrà a creare sarà molto grande.
Per tutte queste ragioni ho valutato i vantaggi di questa scelta superiori agli svantaggi.

Creazione della rete bipartita

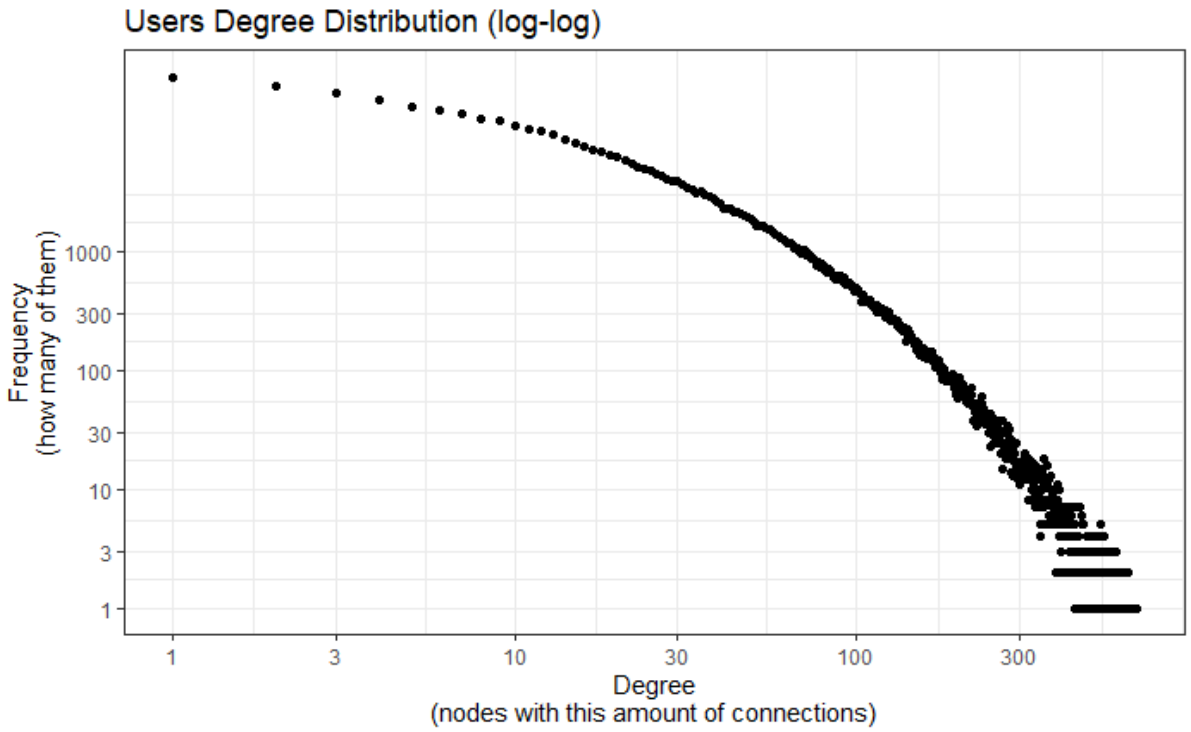
Come anticipato al punto precedente, il punto di partenza della nostra ricerca sarà quello di creare una rete bipartita dove:

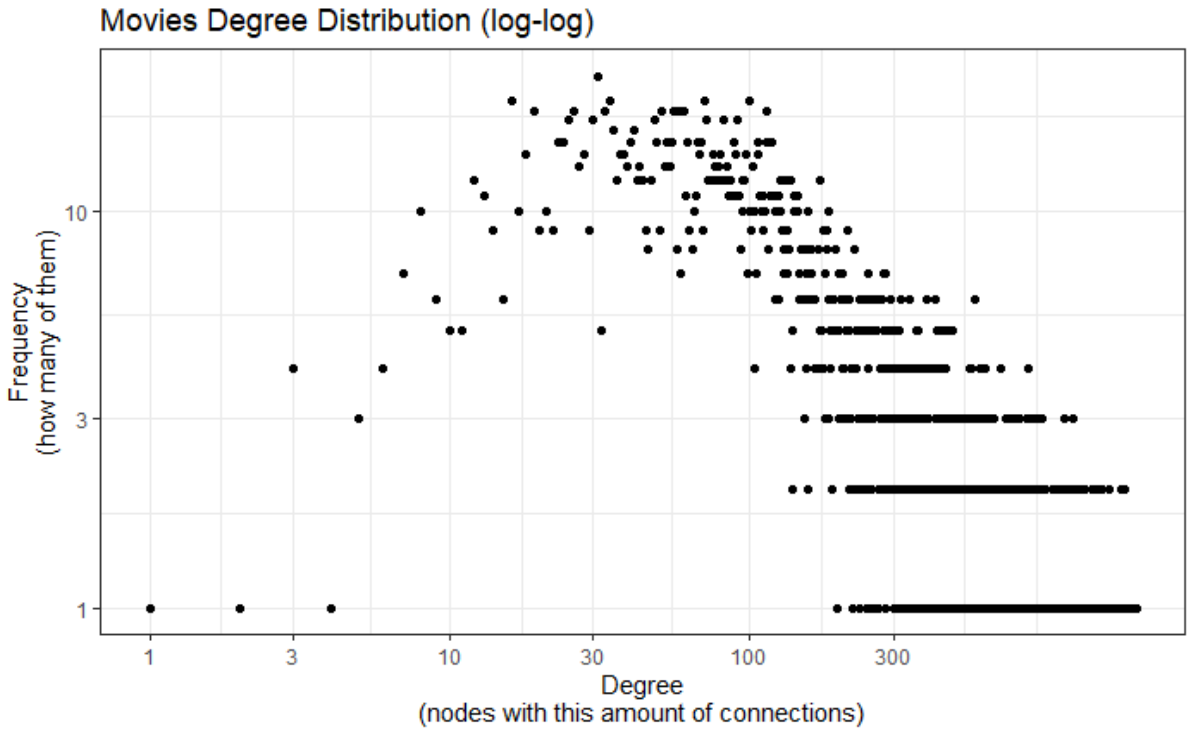
- i due tipi di nodo sono user e movie
- un arco ha un peso pari alla recensione dell'utente al film

Per permettere la visualizzazione del grafo è necessario estrarre un campione di utenti e un campione di film (sia per questioni di limiti fisici del computer, sia perché una rete con più nodi e archi non sarebbe comprensibile).



Si possono analizzare in dettaglio anche la degree distribution di film e utenti, distribuzioni che ho deciso di plottare in scala logaritmica per una migliore visualizzazione dopo alcuni esperimenti.
Si noti che per questa visualizzazione non è stato invece necessario usare solo un campione, in quanto molto più efficiente a livello computazionale.



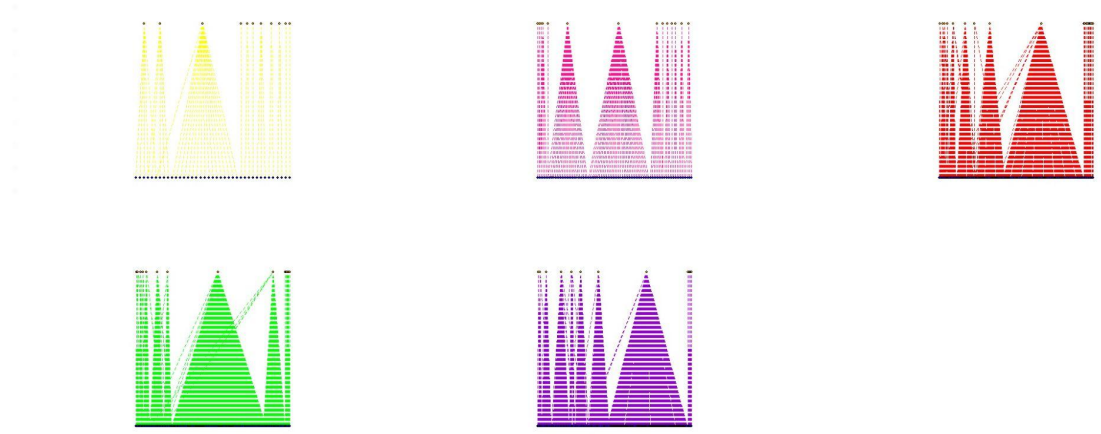


Si può notare che mentre per gli utenti il grafico ha un andamento definito decrescente, per i film, il grafico ha un andamento a parabola dove la funzione sale e poi scende. I nodi più frequenti hanno un grado intermedio.
Queste distribuzioni hanno senso perché la maggioranza degli utenti avrà visto pochi film, mentre pochissimi ne avranno visti centinaia o migliaia (alcuni di quegli utenti potrebbero essere addirittura dei bot). Al contrario la maggior parte

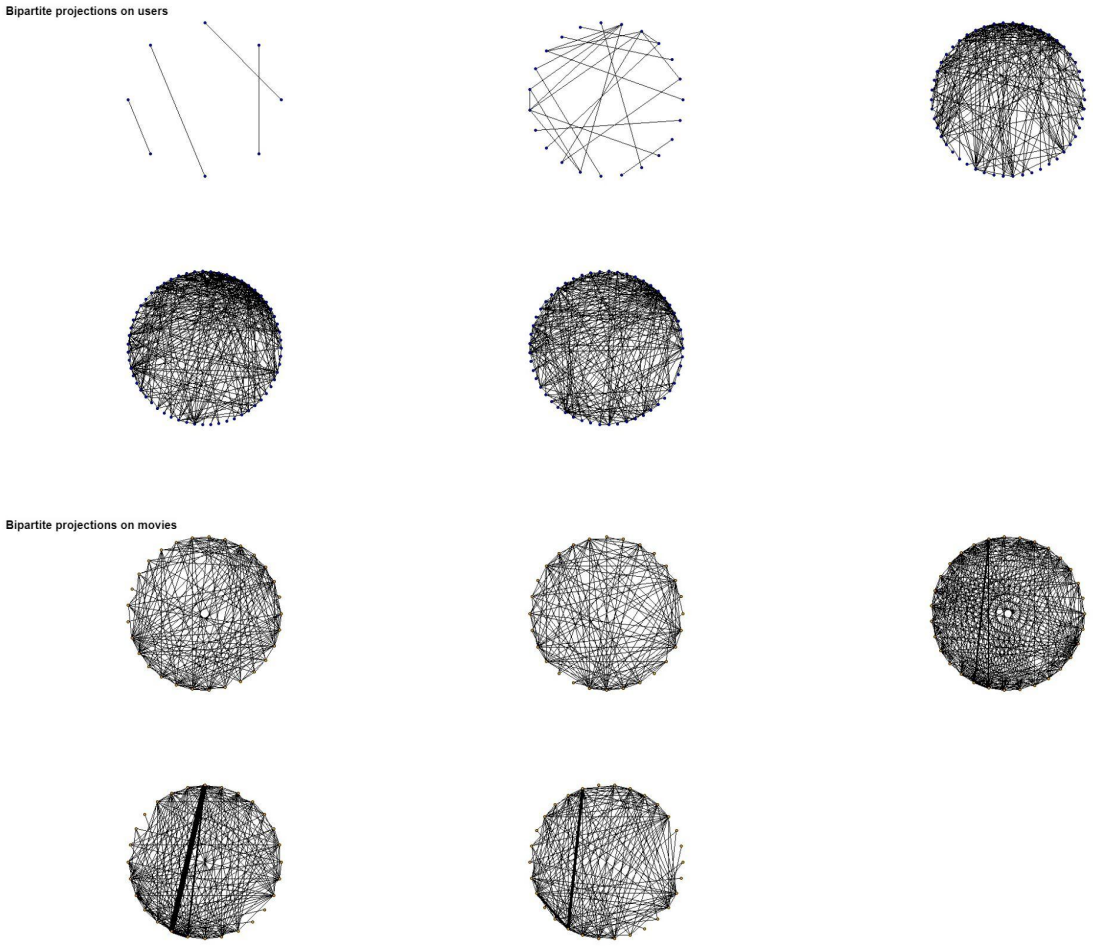
dei film sarà stata vista almeno da un certo numero di utenti, con qualche eccezione (di qualche capolavoro visto da molti o di flop che Netflix ha acquistato visto da pochissimi).

A questo punto noi vorremmo fare la proiezione bipartita della rete tenendo però in considerazione le recensioni (non vogliamo che nella proiezione 2 utenti abbiano un arco in comune perché uno di loro ha messo 5 stelle a un certo film, mentre l'altro ha messo 1 stella allo stesso film).

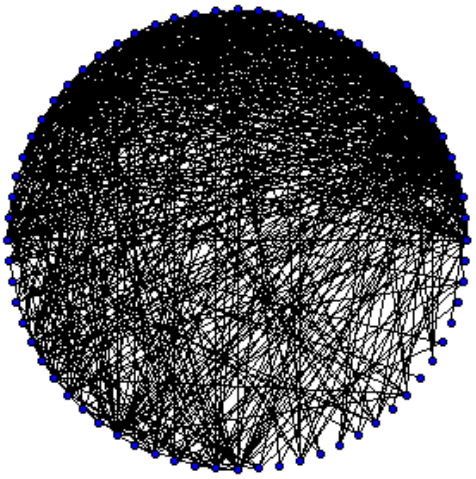
Quindi la prima cosa da fare è dividere il grafo in 5 sottografi indotti dagli archi di peso rispettivamente 1, 2, 3, 4, o 5. In questo modo ciascun sottografo sarà formato da un solo tipo di recensioni.

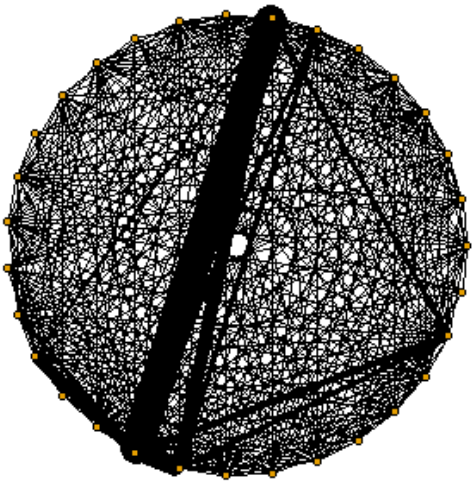


A questo punto è quindi possibile effettuare la proiezione bipartita di tutti e 5 i sottografi sia sugli utenti che sui film (nota: di nuovo per questioni di limiti fisici del computer utilizzato è stato necessario usare un random sample, rispettivamente degli utenti nel primo caso e dei film nel secondo).

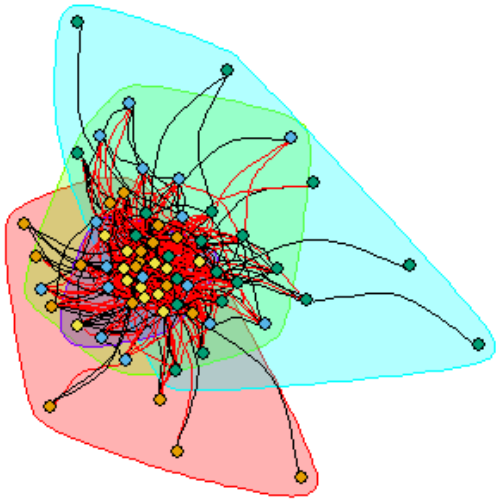


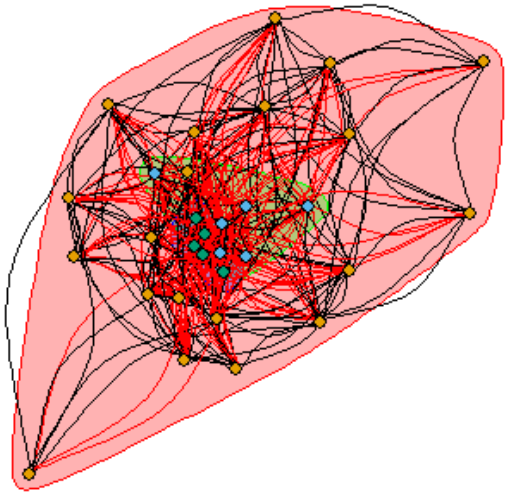
Le 5 proiezioni bipartite possono poi essere rimesse insieme per formare un grafo unico.
Seguono le immagini di proiezioni unite rispettivamente su users e su movies.





Otteniamo così 2 proiezioni bipartite uniche, una per i film e una per gli utenti, su cui possiamo usare algoritmi di clustering per studiarne la struttura e identificare delle comunità.





Queste comunità sono molto particolari, perché, a differenza di quelle che si osservano nella maggioranza delle reti sociali non prevedono alcuna forma di comunicazione tra i loro componenti e permettono quindi di osservare una forma di omofilia incontaminata dall'influenza sociale (almeno all'interno della rete stessa).

Un'ulteriore caratteristica degna di nota è che per come è stata strutturata la rete se due utenti sono in una stessa comunità significa che hanno dato la stessa recensione a dei film tra di loro o a altri membri della comunità e avranno quindi gusti simili.
Allo stesso modo 2 film in uno stesso cluster tenderanno ad avere recensioni simili e quindi è probabile che se ad un utente piace uno dei due film piacerà anche l'altro (o viceversa).
Quindi questi clustering possono essere usati come punti di partenza per costruire dei neighborhood per algoritmi di collaborative filtering come quello usato per vincere la competizione The NetflixPrize.

Data Preprocessing 2

Purtroppo Netflix non ci fornisce alcuna informazione sugli utenti in questione, tuttavia ci fornisce un ulteriore file *movie_title.csv* dove ogni riga è una tripla <*movie_id*, *title*, *year_of_release*> che ci permette quindi di avere 2 nuove informazioni su ciascun film: il titolo e l'anno in cui è stato rilasciato.

Partendo da queste informazioni ho triangolato dove possibile il genere del film utilizzando l'API Cinemagoer di IMdb.
Questa parte è molto onerosa in termini di tempi computazionali a causa della grande dimensione del dataframe e della lentezza intrinseca delle API.

Questo lavoro è l'unica parte del progetto svolta in Python3, il codice è stato scritto sul Jupyter notebook *get_movies_genre.ipynb*.

Per semplicità quindi in questo caso anche la parte di lettura del file *movie_title.csv* è stata fatta in Python.
Dal dataframe processato è stato estratto un campione di movie tra quelli da cui sono riuscito a triangolare il genere con successo, genere cinematografico che è stato poi aggiunto al dataframe in una colonna nominata *genre*.
Il dataframe è stato infine esportato in un file csv denominato *sample_movie_titles.csv*.

Omofilia

E' ora possibile ritornare su R e leggere il dataframe preparato in Python al punto precedente.
Con le nuove informazioni ottenute (anno di rilascio dei film e genere cinematografico), possiamo testare le misure di omofilia dei film sulla rete bipartita.
In questo contesto, quando parliamo di 'omofilia' tra film stiamo implicitamente ponendo la domanda se gli utenti tendono a dare recensioni simili a film prodotti in anni vicini o a film dello stesso genere.

Iniziamo provando a rispondere alla prima domanda: gli utenti tendono a dare recensioni simili a film prodotti in anni vicini?
La prima cosa da fare per rispondere a questa domanda è calcolare il valore atteso del valore assoluto della differenza tra gli anni di rilascio di 2 film connessi da un arco se il nostro grafo fosse un grafo random.
Il valore trovato in questo modo è:

E((|u1$year - u2$year|) | G.israndom) = 20.66026
(u1, u2) ∈ E

Poi troviamo l'effettiva media dei valori assoluti delle differenze tra gli anni di rilascio di 2 film connessi da un arco nel nostro grafo.
Il valore trovato in questo modo è:

mean(|u1$year - u2$year|) = 19.72623
(u1, u2) ∈ E

La differenza media tra anni di film vicini tra loro è effettivamente leggermente inferiore rispetto al valore atteso teorico per grafi random ma non di molto, quindi ha senso andare a cercare ulteriori evidenze.

Utilizzando la clustering membership trovata in precedenza possiamo cercare di capire se i film che si trovano in uno stesso cluster hanno anni di rilascio più vicini rispetto a film che non si trovano nello stesso cluster.
Per questo, consideriamo prima tutti i film (indipendentemente dal cluster di appartenenza) e studiamo media e varianza del loro anno di rilascio

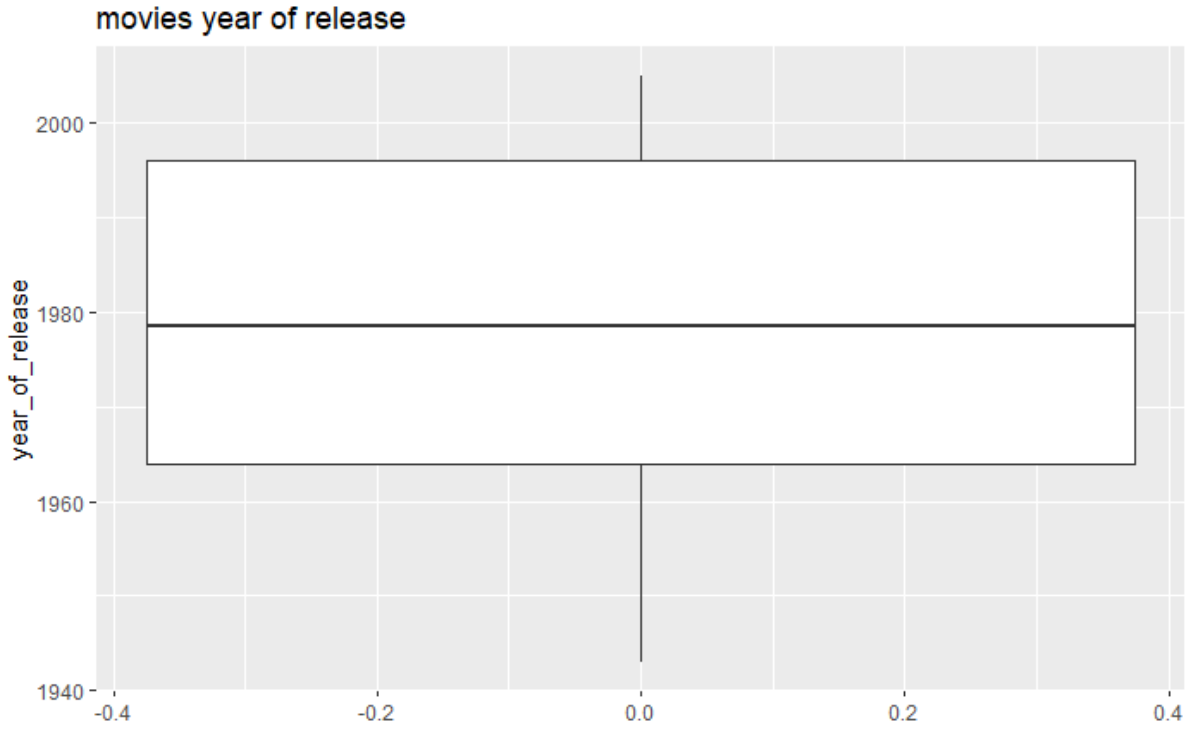
E(u$year) = 1978.875
Var(u$year) = 327.9071
u ∈ V

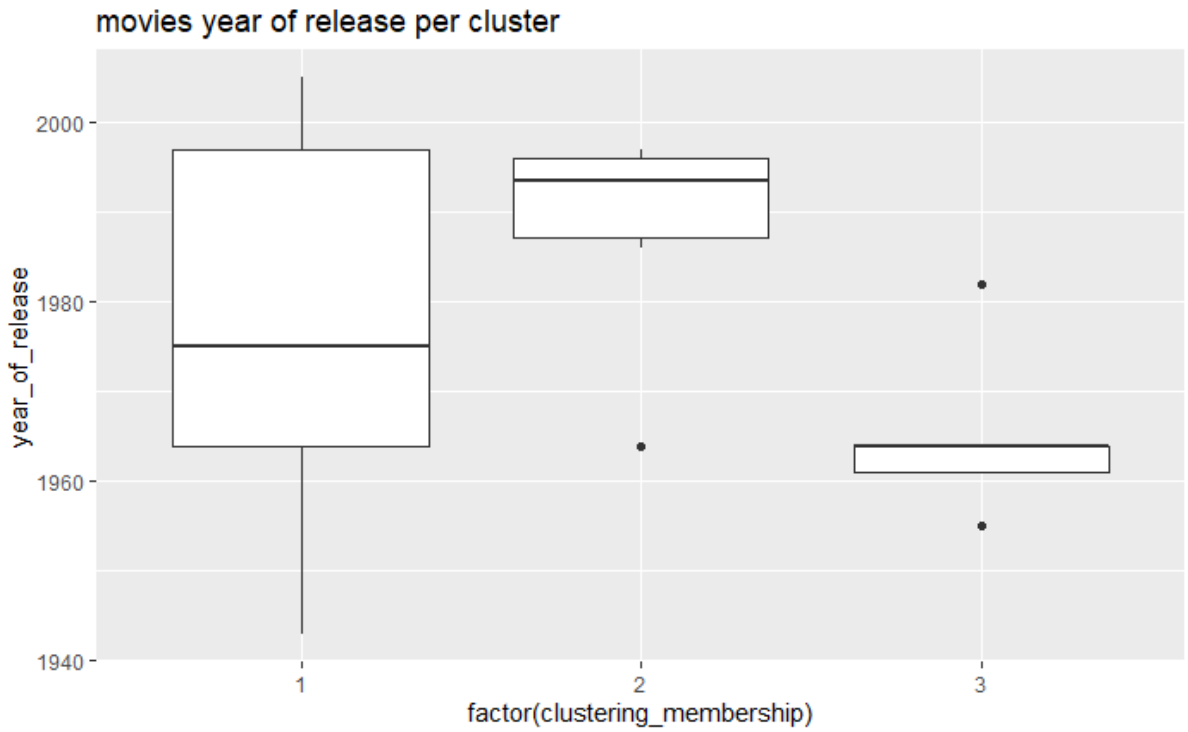
Dopodiché ripetiamo le stesse misure per i film di ogni cluster

E(u1$year) = 1979.276

Var(u1$year) = 360.9926
u1 ∈ V1
E(u2$year) = 1988.333
Var(u2$year) = 159.4667
u2 ∈ V2
E(u3$year) = 1965.200
Var(u3$year) = 101.7000
u3 ∈ V3

Osserviamo anche i boxplot della popolazione intera e delle 3 sottopopolazioni





Solo i film del primo cluster non presentano alcun segno di omofilia, questo ci fa pensare che ci siano alcuni film dello stesso periodo che tendono a ricevere le stesse recensioni dagli utenti (probabilmente fan di quel gruppo di film), mentre molti altri film non presentano segni di omofilia basata sull'anno di rilascio.

Ora proviamo a porci la stessa domanda ma per il genere cinematografico, ovvero: gli utenti tendono a dare recensioni simili a film dello stesso genere?
Il primo step è molto simile a quello svolto per gli anni: determiniamo la probabilità che un arco preso a caso collegherebbe 2 film dello stesso genere cinematografico se il grafo fosse random.
Il valore trovato in questo modo:

$$P(u1\$genre == u2\$genre \mid G.israndom) = 0.1425$$
$$(u1, u2) \in E$$

Ora determiniamo invece la probabilità effettiva che un arco preso a caso colleghi 2 film dello stesso genere cinematografico in questo grafo.
Il valore trovato in questo modo:

$$P(u1\$genre == u2\$genre) = 0.1459016$$
$$(u1, u2) \in E$$

Confrontando i due valori, notiamo che la probabilità effettiva è pressoché identica a quella teorica.
E' quindi molto probabile che non vi sia alcuna omofilia per genere cinematografico.

Per confermare la nostra ipotesi, svolgiamo un'ulteriore analisi basata sulla clustering membership dei nodi.
Vogliamo vedere se la probabilità di 2 nodi presi a caso di avere lo stesso genere cinematografico aumenta se i 2 nodi possono essere estratti solo all'interno di uno stesso cluster.
I valori così trovati sono per ciascun cluster rispettivamente:

P(u1$genre == u2$genre) = 0.14532020
(u1, u2) ∈ E1
P(u1$genre == u2$genre) = 0.06666667
(u1, u2) ∈ E2
P(u1$genre == u2$genre) = 0.20000000
(u1, u2) ∈ E3

E la media di queste probabilità è

mean(P) = 0.137329

L'unico cluster a presentare una lieve evidenza di omofilia è il cluster 3, tuttavia la probabilità media di 2 nodi presi a caso di avere lo stesso genere cinematografico dove i 2 nodi possono essere estratti solo all'interno di uno stesso cluster, è addirittura inferiore rispetto a quella del grafo random. Questo risultato, unito a quello ottenuto al punto precedente ci lascia intendere che non c'è nessuna omofilia evidente per genere tra i film.

Conclusione

In conclusione possiamo quindi dire che ci sono dei gruppi di film che sono usciti in anni vicini e tendono ad avere recensioni simili dagli stessi utenti che evidenzia con ogni probabilità delle fasce di utenti appassionati di film di determinati periodi.
Al contrario, le nostre analisi non hanno invece portato ad alcuna evidenza per quanto riguarda il genere cinematografico. Non sembra che il genere cinematografico sia un fattore discriminante per quanto riguarda ricevere recensioni simili dagli stessi utenti.
Come sempre in questi casi, le misure matematiche sono esatte ma l'interpretazione e le conclusioni tratte presentano una componente intrinseca e inscindibile di soggettività.

Bibliografia

- Note lezioni di Social Network Analysis 2021-21 della professoressa Zollo F.
- Note lezioni di Social Network Analysis 2021-21 del dottor Galeazzi A.
- Networks, crowds, and markets. David Easley and Jon Kleinberg, Cambridge University Press, 2012 (complete pre-publication draft of the book: https://www.cs.cornell.edu/home/kleinber/networks-book/)
- Cinemagoer (https://cinemagoer.readthedocs.io/en/latest/index.html) API for IMdb (https://www.imdb.com/ https://developer.imdb.com/)
- Kaggle (https://www.kaggle.com/netflix-inc/netflix-prize-data?select=probe.txt)
- Netflix for the data release for The NetflixPrize competition (https://www.netflix.com)
- Presentation fonts (http://pptmon.com/)

Note

Per eseguire il progetto, assicurarsi di sostituire la variabile path con il percorso con il quale si vuole eseguire e scaricare tutte le librerie usate.
Il primo chunk del file principale del progetto: *SNAProjectNetflixPrize.Rmd* contiene le istruzioni

rm(list = ls())
set.seed(76418)

così, grazie alla prima istruzione, ogni volta che viene eseguito il file RMarkdown, vengono rimosse automaticamente tutte le variabili globali, tuttavia, grazie alla seconda istruzione, il codice si ripete sempre allo stesso modo.
Il seme scelto è stato ottenuto eseguendo un'istruzione per ottenere un intero causale, in questo modo è garantita la randomicità dell'esecuzione evitando ogni bias umano.
Questo progetto nella sua interezza è stato sviluppato per soli fini accademici.
Il materiale consegnato verrà mantenuto in una cartella drive dove potrà essere consultato nella versione corretta per evitare problemi dovuti a sformattazione: https://drive.google.com/drive/folders/1628c4VTYPqF7o50WOqI6btamX2XgfsEV?usp=sharing