

ABSONES: Agent Based SOcial NEtwork Simulator

Simone Ciccolella
s.ciccolella@campus.unimib.it
Mat. 762234

Daniele Bellani
d.bellani1@campus.unimib.it
Mat. 780675

Sommario. In questo progetto proponiamo un modello multi-agente per la simulazione di social network, in particolare la piattaforma di microblogging Twitter. Il nostro approccio prevede la rappresentazione del sistema come un grafo diretto e l'identificazione degli utenti come agenti di tale sistema complesso, che si attivano e compiono azioni sulla base di probabilità ed estrazioni di tipo Monte Carlo. Gli obiettivi che ci siamo posti sono stati la prevenzione del *completamento* della rete e il mantenimento del principio di *invarianza di scala*, così da simulare, in modo quanto più fedele, il comportamento della vera rete sociale. Con il modello così costruito, è possibile studiare fenomeni sociologici usando scenari simulati nel modo più attinente possibile alla realtà. In particolare, abbiamo provato a simulare il “salto” di popolarità di un individuo poco conosciuto in seguito all’interazione con un utente più “in vista”, traendo ispirazione da quanto avvenuto nel 2012 tra l’On. Maurizio Gasparri e l’utente Daniele Termite.

Indice

1	Introduzione	3
2	Letteratura correlata	4
3	Riferimenti matematici	7
3.1	Cosine similarity	7
3.2	Teoria dei grafi	7
3.2.1	Coefficiente di clustering	7
3.2.2	Assortatività	8
3.2.3	Rete a invarianza di scala	8
4	Descrizione modello	9
4.1	Utenti	10
4.2	Attività di base	11
4.3	Attività sociali	12
4.4	Step Simulazione	14
5	Interpretazione multi-agente	14
6	Eperimenti	16
6.1	Fase 1	17
6.1.1	Risultati	17
6.1.2	Commento	27
6.2	Fase 2	27
6.2.1	Risultati	28
6.2.2	Commento	32
7	Conclusione e sviluppi futuri	32
8	Disponibilitá software	33

1 Introduzione

I social network sono uno strumento che è diventato parte integrante della vita di tutti. Nel giugno 2017 la piattaforma Facebook ha passato la soglia dei due miliardi di utenti attivi su base mensile [1]; nel frattempo, Twitter si è imposto come mezzo di comunicazione principale tra i cosiddetti “influencer”, con un grosso impatto sull’opinione pubblica (per esempio, giocando un ruolo fondamentale nell’elezione di Donald Trump [2]). L’aspetto principale di questa forma di comunicazione è la produzione, da parte degli utenti, di un’enorme mole di dati: le acquisizioni di YouTube da parte di Google [3] e di LinkedIn da parte di Microsoft [4] (e le cifre in gioco) testimoniano l’interesse delle grandi aziende per queste sorgenti di dati. L’analisi di questi ultimi aiuta a determinare strategie di mercato, a personalizzare raccomandazioni di prodotti, studiare e prevedere il sentimento su un evento o un prodotto, oppure condurre studi di stampo sociologico.

Sorgono però alcune difficoltà. In primo luogo, raramente questi dati vengono rilasciati in formato aperto (Open Data). Inoltre, le loro enormi proporzioni ne rendono difficile l’analisi e la gestione. Da qui la necessità di svolgere delle simulazioni *in-silico* con modelli sviluppati in modo da essere il più possibile veritieri. L’attendibilità di queste simulazioni dipende dal modello su cui vengono eseguite e dalle inevitabili assunzioni che sono state fatte durante la sua costruzione. L’obiettivo è quindi produrre sistemi artificiali che siano replicate fedeli di sistemi complessi reali. Una possibile strada è quella dei *sistemi multi-agente*. Questi si basano sulla definizione di *agente* (vedi sezione 5), un entità virtuale [8] capace di compiere azioni ed interagire con altri agenti all’interno di un ambiente (*environment*). Si può ottenere una rappresentazione multi-agente di una rete sociale rappresentando l’ambiente come un grafo (connesso o non connesso) e gli agenti come nodi di questo grafo. Gli archi rappresentano le interazioni tra i vari agenti, siano esse relazioni di “amicizia” (es. Facebook), oppure di “subscription” (es. Twitter). Il tempo viene spesso rappresentato in modo discreto dalle *iterazioni* o *step*. Ad ogni step, i nodi decidono individualmente le azioni da compiere, per esempio se stabilire o meno un nuovo collegamento con un altro nodo. Ciò rende il sistema *dinamico* ed in continua evoluzione. Un modello è considerato attendibile se conserva le caratteristiche proprie di una rete sociale per tutta la durata della simulazione. In questo modo è possibile simulare scenari utili alla previsione di fenomeni sociali nel modo più fedele possibile.

Negli ultimi anni sono stati proposti alcuni modelli per la simulazione multi-agente di social network, con risultati alterni e non definitivi, dovuti alla complessità del tema. Da questi abbiamo tratto ispirazione per il nostro modello. Nella prima sezione esamineremo alcune delle metodologie proposte; successivamente, dopo aver fissato alcune definizioni di carattere matematico utili per le sezioni seguenti, passeremo all’illustrazione della nostra soluzione e della visione agent-based dello stesso. Nelle ultime due sezioni esporremo gli esperimenti condotti e ne commenteremo i risultati.

2 Letteratura correlata

La letteratura riguardante modellazione multi-agente di social network è relativamente scarsa. Nonostante la pubblicazione in formato open di dati provenienti da alcune piattaforme, risulta comunque molto difficile avere una visione generale di ciò che accade in sistemi così ampi e complessi. Una situazione migliore si riscontra nella letteratura riguardante l'analisi a posteriori di queste reti, da tempo molto studiate. Non mancano quindi metriche e parametri per la valutazione di reti sociali, anche se le visioni in merito sono molte e a volte piuttosto discordanti.

Un primo lavoro degno di nota è quello pubblicato da Hamill e Gilbert [5]. In primo luogo, gli autori stabiliscono quali sono le caratteristiche che una rete sociale simulata dovrebbe avere, tra cui le più importanti sono:

Bassa densità di rete La densità di una rete [5] (*network density*) è definita come il rapporto tra il numero di archi esistenti e il numero massimo di archi possibili. Un utente medio è collegato con un numero di utenti dell'ordine delle centinaia o poche migliaia, numero che, se confrontato con le centinaia di milioni di utenti (se non miliardi) di tutto il sistema, risulta essere piuttosto basso.

Assortatività positiva Con questo termine, gli autori indicano la tendenza dei nodi con più connessioni ad essere collegati con altri nodi molto connessi (vedi 3.2).

Presenza di comunità Ovvero, la tendenza a formare *clusters*, gruppi di nodi fortemente connessi tra di loro ma debolmente connessi con il resto del sistema. Viene introdotto, a questo proposito, il *coefficiente di clustering* (*clustering coefficient*, vedi 3.2)

Lunghezza ridotta dei cammini Secondo gli autori, in media si può raggiungere un utente di un social network partendo da un qualsiasi altro nodo compiendo solo pochi passi, ovvero percorrendo un cammino ridotto. La lunghezza dipende dalle proporzioni della rete. Questo è un effetto molto noto in letteratura, e prende il nome di *small-world effect* [6][18].

Vengono esposti inoltre diversi tipi di rete, emersi nel corso degli anni in letteratura (vedi fig. 1):

Regular lattice Ogni nodo è collegato ad un numero fisso di suoi vicini

Random network Ogni nodo è collegato in media ad un certo numero di altri nodi

Small world network Basato sul modello *regular lattice*, aggiunge o riarrangia collegamenti in modo casuale

Scale-free network Descritta per la prima volta da Barabási & Bonabeau [7], prevede che pochi nodi abbiano molti collegamenti (vedi sezione 3.2).

Gli autori indicano la costruzione *scale-free* come la migliore tra le quattro, in quanto presenta tutte le caratteristiche elencate in precedenza, con l'eccezione dell'*assortatività*, non particolarmente riflessa nel modello. Passano quindi all'esposizione della loro proposta, un modello ad agenti basato sul concetto di *social circles* [5]: ogni agente può stabilire un “link” con un altro agente solo se quest'ultimo può fare altrettanto. Quest'idea di *reciprocità* si adatta bene alla modellazione di alcune piattaforme (es. Facebook), mentre si adatta meno su altre: un esempio è Twitter, dove per stabilire un collegamento, il fatto che due utenti si conoscano direttamente è poco rilevante.

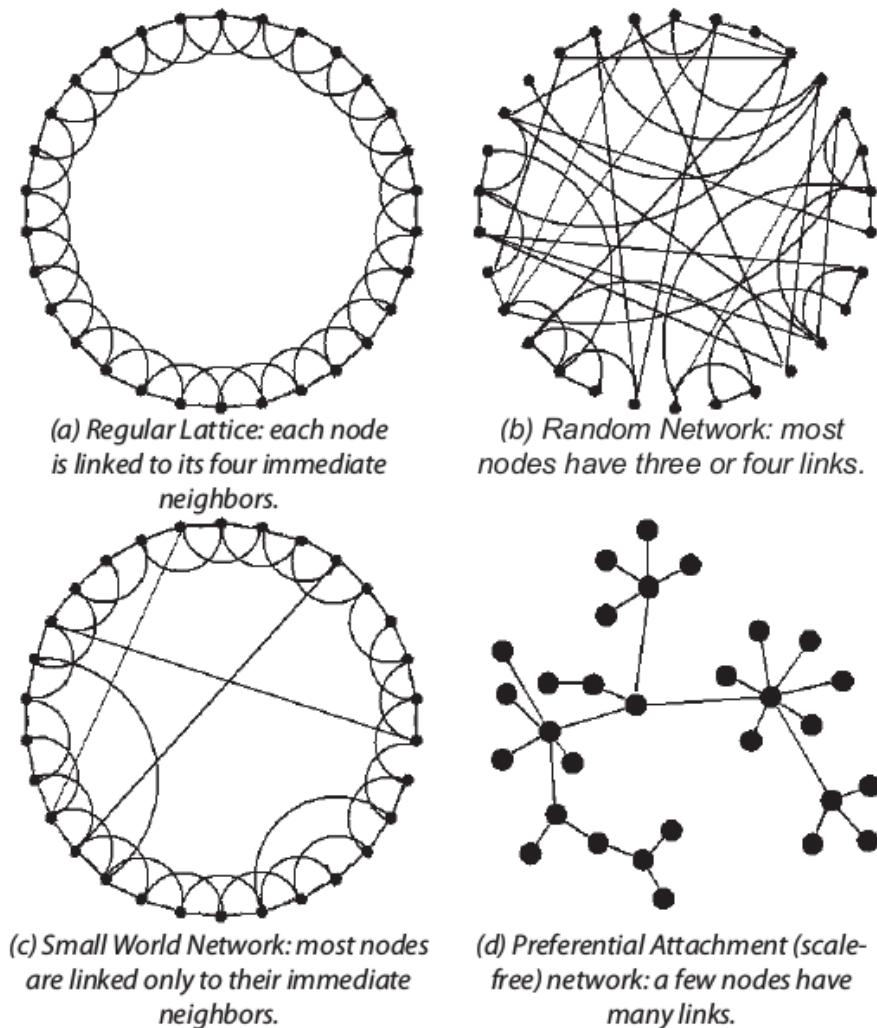


Figura 1: I quattro modelli di rete descritti in [5]

L’idea che una rete sociale sia assimilabile a una rete ad invarianza di scala (ovvero i cui nodi seguono una *power law*, vedi 3.2) è largamente supportata in letteratura: testi importanti e fondamentali come [6] e [18] asseriscono che i social network, così come il World Wide Web stesso, siano organizzati in modo tale che pochi nodi abbiano molti collegamenti e siano collegati tra di loro, mentre molti nodi abbiano pochi collegamenti e siano collegati ad alcuni di questi “hub”. Entrambi i testi, così come [5], ricordano però l’assunzione *small-world*, che, specialmente per i social network, debba essere verificata.

Un altro lavoro che abbiamo recuperato è quello di Liben-Nowell & Kleinberg [9]. Il problema da loro affrontato è quello della previsione di collegamenti (*link prediction*) nei social network, ovvero prevedere, dato lo stato del sistema in un dato istante di tempo, la formazione di nuovi collegamenti negli istanti immediatamente successivi. In una simulazione *dinamica* di una rete è fondamentale stabilire un metodo per cui nuovi collegamenti vengono stabiliti tra i nodi. Un aspetto cruciale in questo caso è il criterio con cui viene valutata la *similarità* tra i nodi del grafo, ovvero come viene assegnato uno *score* a una coppia di nodi (arco) in modo che ne misuri la “*distanza*” rispetto ad una particolare proprietà o caratteristica. Nodi simili ma non ancora collegati avranno infatti un’alta probabilità di stabilire un collegamento negli istanti di tempo più prossimi.

Nell’articolo viene fatta una rassegna dei metodi principali; l’obiettivo di questi metodi è costruire la matrice di similarità :

Neighborhood-based methods Di questa categoria fanno parte tutti quei metodi che si basano sul *vicinato* (*neighborhood*) dei nodi di cui si vuole calcolare la similarità. La misura di distanza può essere il numero di nodi adiacenti in comune, oppure la probabilità che i due nodi in esame abbiano un vicino in comune, per esempio usando la *distanza di Jaccard*. In alternativa, questa probabilità può essere calcolata in modo proporzionale alla dimensione dei vicinati dei due nodi candidati, per esempio moltiplicandone la cardinalità.

Paths-based methods Anche la distanza intesa come lunghezza di un cammino tra due nodi può essere intesa come misura di similarità; per esempio, in [10] viene descritta una metrica che prende in considerazione la lunghezza tutti i cammini esistenti tra due nodi per quantificarne la similarità. Anche alcuni celebri algoritmi, come il *PageRank* di Google [11], fanno parte di questa categoria.

Altre tecniche possono essere usate in congiunzione con i metodi sopra elencati, per semplificare la computazione oppure per irrobustirne la previsione. Procedure di *clustering* o *matrix factorization* possono aiutare ad eliminare i collegamenti meno significativi prima dell'effettivo calcolo delle metriche.

Essendo però il nostro obiettivo la simulazione di una *social network*, non si può ignorare la (forte) componente sociale del sistema considerato. Ogni nodo rappresenta un utente, e dunque ne eredita le caratteristiche personali come interessi e carattere. La letteratura specializzata in *social sciences* ha da tempo definito il concetto di *omofilia* (*homophily*) [12] [18], inteso come la tendenza di ogni individuo a stringere legami con altri individui dalle caratteristiche simili. Nel computo della misura di similarità (o dissimilarità) bisognerebbe quindi tenere conto di tali caratteristiche e trovare quindi il modo di rappresentarle nel sistema. In [13], per esempio, gli autori hanno tentato di tradurre in linguaggio matematico l'idea di omofilia, in particolare per quanto riguarda l'aggregazione di individui in base al gruppo di appartenenza. Dopo aver effettuato un'etichettatura degli individui in base a questi raggruppamenti, hanno definito un indice H_i per l'omofilia:

$$H_i = \frac{s_i}{s_i + d_i} \quad (1)$$

dove s_i indica il numero medio di *amicizie* (*friendships*) che un individuo facente parte del gruppo i ha con i membri dello stesso gruppo, mentre d_i indica il numero medio di *amicizie* che un membro del gruppo i ha con gli individui appartenenti ad altri gruppi. Questa misura, se confrontata con la frequenza relativa w_i degli individui appartenenti al gruppo i , fornisce un'indicazione del comportamento degli agenti all'interno del sistema, ovvero se tendono a stabilire collegamenti fra individui dello stesso gruppo (*inbreeding homophily*) oppure con individui appartenenti ad altri gruppi (*heterophily*).

Occorre notare come, nell'ambito dei social network, il termine *omofilia* e il termine *assortatività* siano spesso indicati come sinonimi. [20] e [21] per esempio, forniscono rispettivamente una misura di assortatività per attributo e per grado (vedi 3.2), ovvero un indice di quanto i vertici di una rete siano legati a nodi con simili valori per l'attributo considerato oppure con simili gradi di archi entranti o uscenti.

Esistono modelli agent-based oltre a quello esposto in [5], per esempio quelli descritti in [14] e [15]; entrambi usano una rete *scale-free* come ambiente per gli agenti (il primo orientata, la seconda non orientata), ed entrambi regolano gli eventi che accadono nel sistema e le azioni degli agenti secondo degli spazi di probabilità, che si aggiornano dinamicamente. Queste probabilità sono in funzione dell'*in-degree* di ogni nodo e della *similarità* tra ogni coppia di nodi. In particolare, [14] contempla anche l'aggiunta e la delezione di nodi durante la simulazione, operazioni anch'esse regolate da una specifica distribuzione di probabilità.

Un’ultima risorsa da noi sfruttata è stata la relazione stilata da Marco Comi e Marco Gravina che, come noi e prima di noi, si sono cimentati nella simulazione del social network Twitter. Abbiamo preso spunto dalle loro soluzioni e dalle criticità del loro modello per sviluppare la nostra proposta.

3 Riferimenti matematici

Di seguito descriviamo brevemente alcuni concetti matematici fondamentali per la comprensione delle sezioni successive.

3.1 Cosine similarity

Il *coseno di similitudine* (*cosine similarity*) [16] è una misura di similarità tra due vettori v_1 e v_2 , definita in questo modo:

$$\text{sim}(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|} \quad (2)$$

dove il numeratore rappresenta il *prodotto scalare* tra i due vettori, mentre il denominatore rappresenta il prodotto dei moduli. I valori possibili ricadono nell’intervallo $[-1, 1]$, dove 1 si ottiene in caso di vettori identici, mentre -1 in caso di completa dissimilarità (vettori opposti).

3.2 Teoria dei grafi

Un *grafo orientato* G [17] è una coppia (V, E) , dove V (insieme dei *vertici*) è un insieme finito ed E è una relazione binaria in V .

Se (u, v) è un arco [17] di un grafo $G = (V, E)$ diciamo che il vertice v è *adiacente* al vertice u . Dato un grafo G orientato [17], il *grado uscente* (*out-degree*) di un vertice è il numero di archi che escono dal vertice; il *grado entrante* (*in-degree*) è il numero i archi che entrano nel vertice. Un cammino (*path*) [6] da un vertice v_0 a un vertice v_n è una lista ordinata di archi $P = \{(v_0, v_1), (v_1, v_2), \dots, (v_{n-1}, v_n)\}$, e n corrisponde alla lunghezza di questo cammino.

Un grafo orientato G si dice *completo* [6] quando ogni coppia di vertici è collegata da una coppia simmetrica di archi. La definizione è analoga al caso in cui il grafo sia non orientato, con la differenza che, in quest’ultimo, ogni coppia di archi opposti situata tra due nodi è sostituita da un solo arco non orientato. Il numero di archi [6] in un grafo *non orientato completo* è pari a $\frac{N(N-1)}{2}$, dove N è il numero di vertici del grafo. Se si escludono i *cappi* (*self-loops*), allora un grafo *orientato completo* è composto da $N(N - 1)$ archi.

Si dice *densità della rete* il rapporto tra il numero di archi esistenti e il numero di archi possibili (es. un grafo con densità al 50% sarà composto da un numero di archi che è pari alla metà del totale di archi possibili).

3.2.1 Coefficiente di clustering

Il *coefficiente di clustering* (*clustering coefficient*) [6], in un grafo *non orientato*, cerca di esprimere il grado con cui i vicini di un dato nodo sono collegati tra di loro. Questo fornisce un’idea della compattezza del gruppo in cui è inserito quello specifico nodo. È possibile inoltre avere una misura globale della tendenza dei nodi a “raggrupparsi” calcolando il *coefficiente di clustering medio* (*average clustering coefficient*). Nei grafi orientati [18] è disponibile una misura

analoga, chiamata *coefficiente di clustering globale* (*global clustering coefficient*, a volte anche chiamata *ratio of transitive triplets* [6]); la sua definizione è la seguente:

$$C_\Delta = \frac{3 \times \text{NumeroDiTriangoli}}{\text{NumeroDelleTripleConnesse}} \quad (3)$$

dove per *triangolo* si intende un insieme di 3 vertici, ognuno dei quali connesso agli altri due da archi, mentre per *tripla connessa* si intende tre vertici uvw connessi dagli archi (u, v) e (v, w) . I valori possibili ricadono nell'intervallo $[0, 1]$.

3.2.2 Assortatività

Una rete si dice *assortativa* (*assortative*) [18] se una frazione significativa dei suoi archi collegano vertici simili tra di loro. La similitudine può essere calcolata rispetto a un particolare attributo dei nodi, o, in alternativa, rispetto al *grado* dei vertici. In quest'ultimo caso, in un grafo diretto [20] la *in-assortativity* e la *out-assortativity* misurano rispettivamente la tendenza dei nodi a legarsi con altri nodi che hanno *in-degree* o *out-degree* identico al loro. Sia $r()$ la funzione che calcola la assortatività, e siano indicati i due tipi di assortatività appena descritti come $r(in, in)$ e $r(out, out)$, è possibile estendere la portata della definizione definendo anche $r(in, out)$ e $r(out, in)$. Supponendo di voler calcolare $r(in, out)$, si ricorre alla seguente formula [21]

$$r = \frac{\sum_{jk} jk(e_{jk} - q_j^{in}q_k^{out})}{\sigma_{in}\sigma_{out}} \quad (4)$$

dove e_{jk} è la probabilità che un arco qualsiasi conduca da un nodo con *in-degree* j a un nodo con *out-degree* k , σ_{in} è la deviazione standard della distribuzione q^{in} e σ_{out} è la deviazione standard della distribuzione q^{out} . La distribuzione di probabilità q_k^{out} (e analogamente q_j^{in}) è calcolata come segue:

$$q_k^{out} = \frac{(k+1)p_{k+1}^{out}}{\sum_k kp_k^{out}} \quad (5)$$

dove p_k^{out} è la probabilità che un nodo abbia *out-degree* k . Al denominatore si trova l'*out-degree* medio della rete. L'assortatività per attributo è analoga, e si calcola in questo modo:

$$r = \frac{\sum_{xy} xy(e_{xy} - a_x b_y)}{\sigma_a \sigma_b} \quad (6)$$

dove e_{xy} è la probabilità che un arco qualsiasi conduca da un nodo con valore dell'attributo x a un nodo con valore y , mentre a e b sono rispettivamente le frequenze degli archi che escono ed entrano in un nodo con valori x e y . In entrambi i casi il valore di r ricade nell'intervallo $[-1, 1]$, con 1 a indicare la perfetta assortatività e -1 la perfetta disassortatività.

3.2.3 Rete a invarianza di scala

Una rete viene indicata come *rete a invarianza di scala* (*scale free network*) se la distribuzione dei gradi dei nodi (probabilità $p(k_i)$ che un nodo scelto in modo random abbia grado k_i) segue una *power law* [7].

Una *power law* [6] è una funzione $y = f(x)$ in cui il valore y della funzione è proporzionale ad una potenza del valore in ingresso x :

$$p_x \sim x^{-\gamma} \quad (7)$$

L'equazione 7 è chiamata *power law distribution* e l'esponente $-\gamma$ è detto *degree exponent*. La rappresentazione con il logaritmo dell'equazione 7 diventa

$$\log p_x \sim -\gamma \log p \quad (8)$$

Se vale l'equazione 8, allora ci si aspetta che $\log p_x$ dipenda linearmente da $\log x$, con inclinazione della retta pari a γ , come indicato nella figura 2. Assumendo che il grado di un nodo sia una quantità discreta, si può adottare il formalismo per il discreto; in questo modo, l'equazione 7 diventa

$$p_x \sim Cx^{-\gamma} \quad (9)$$

dove C è una costante.

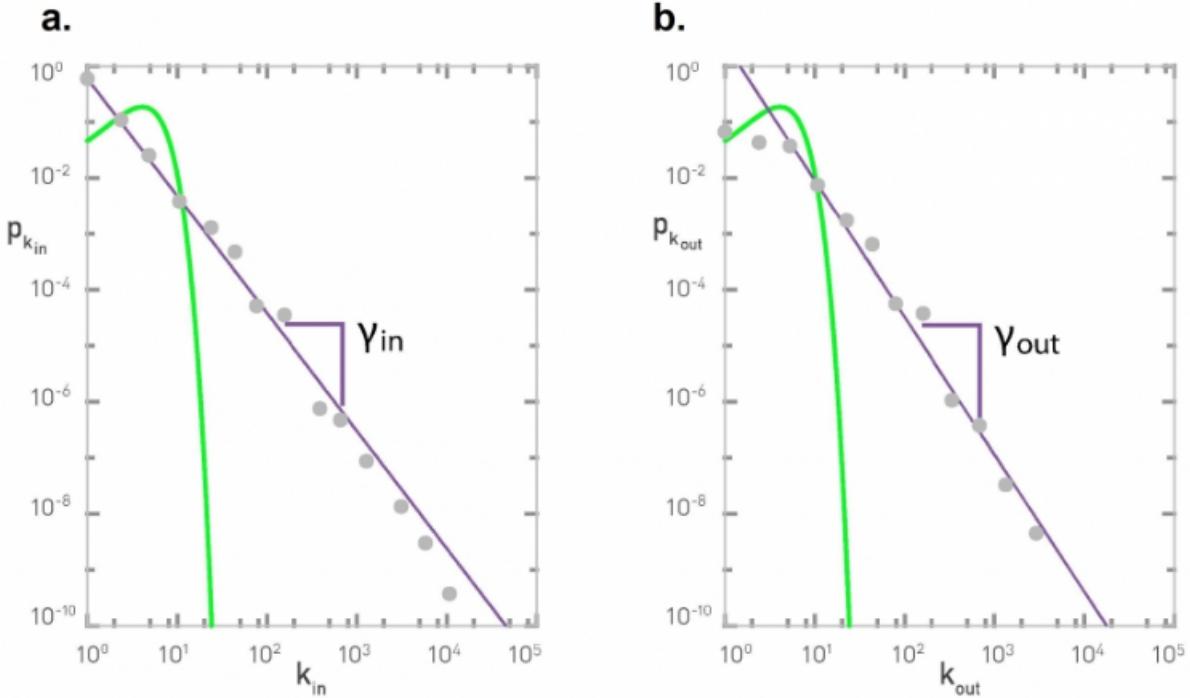


Figura 2: Distribuzione del WWW misurata nel 1999 [6]. La figura a. rappresenta la distribuzione dei nodi rispetto all'*in-degree*, mentre b. rispetto all'*out-degree*. Entrambe usano una scala logaritmica.

4 Descrizione modello

Abbiamo cercato di creare un modello che rispecchi il più possibile il modo in cui Twitter viene utilizzato nella realtà da utenti umani. Ovviamente alcune restrizioni e semplificazioni sono state effettuate per creare un modello realizzabile sia dal punto di vista concettuale che computazionale. Tuttavia ABSoNeS risulta essere una buona base per la simulazione di social network, come verrà descritto più avanti. Alcune delle semplificazioni più rilevanti possono comunque essere introdotte in futuro senza la necessità di stravolgere il modello attuale, ma arricchendolo con relativa semplicità.

ABSoNeS è composto da un *grafo* i cui nodi rappresentano gli *utenti* (quindi gli *agenti*) di una rete sociale ispirata a Twitter. Ogni nodo può produrre oggetti (tweet e retweet) e stabilire o interrompere collegamenti secondo delle *probabilità* (statiche o dinamiche); l'agire o meno di questi utenti è determinato secondo un campionamento di tipo Monte Carlo su queste probabilità. Ogni nodo ha degli attributi che lo rendono unico, dunque ognuno di questi agenti sviluppa un comportamento che lo identifica, ed è parzialmente influenzato dalle azioni degli agenti con cui è in contatto.

Di seguito elenchiamo gli elementi chiave comuni a tutti gli utenti. Tali aspetti servono a descrivere le caratteristiche personali degli utenti, in particolare il loro periodo di attività durante il giorno e i loro interessi personali. Questi elementi sono al momento immutabili dopo la creazione dell'utente, per quanto questa possa sembrare una forte restrizione, non è sbagliato assumere che mediamente gli interessi e gli impegni lavorativi di una persona non cambino nell'arco di qualche mese, che è l'obiettivo della simulazione.

Fasi Temporali (FT) Dividiamo la giornata in 12 fasi che rappresentano 2 ore l'una in cui ogni utente ha un personale valore di *activity* che rappresenta il suo utilizzo di Twitter. Tali fasi sono così suddivise:

- 4 fasi con *activity* pari a 0 corrispondente a 8 ore di sonno, ovvero la quantità consigliata (e quasi mai rispettata);
- 4 fasi con *activity* bassa, corrispondente a 8 ore di lavoro;
- 4 fasi con *activity* elevata che rappresenta 8 ore di tempo libero in cui l'utente ha una attività maggiore su Twitter.

Topic ovvero i possibili argomenti di interesse esistenti nel modello perciò abbiamo una lista $topic = (t_1, \dots, t_n)$ definiti a priori e immutabili. Non vengono mai utilizzati direttamente, ma vengono utilizzati come indici di riferimento dunque non è necessario implementarli realmente.

4.1 Utenti

Gli utenti sono i nodi della rete. Di seguito mostriamo come gli utenti sono rappresentati nel modello, in particolare ogni utente è caratterizzato da:

Personal Interest (PI) è un vettore di probabilità $PI = (p_1, \dots, p_n)$ dove $p_i \in (0, 1)$. indica la probabilità di interesse del nodo rispetto al *topic* i generata casualmente. Da notare che $\sum_{i=1}^n p_i = 1$.

Timezone (TZ) ovvero una lista $TZ = (tz_1, \dots, tz_{12})$ dove i tz_i sono generati in accordo con le **FT** come segue:

- Viene scelto un indice $1 \leq i \leq 12$ casualmente
- Si impostano le ore di sonno $tz_j = 0$ per $j = i, \dots, i + 3$
- Si impostano le ore di lavoro $tz_j = \text{bassa activity}$ per $j = i + 4, \dots, i + 7$
- Si impostano le ore di tempo libero $tz_j = \text{alta activity}$ per $j = i + 8, \dots, i + 11$

Tutte le precedenti operazioni di indici sono da considerarsi mod 12. Il motivo di questa scelta è quello di simulare sia diversi orari per le persone, come ad esempio lavori notturni, sia diversi fusi orari cercando di distribuire uniformemente gli orari personali. Nel codice, i valori per alta e bassa activity sono probabilità, e vengono generate usando una distribuzione triangolare con i seguenti parametri:

- Bassa activity: $a = 0, m = 0.25, b = 0.6$
- Alta activity: $a = 0.4, m = 0.75, b = 1$

Ogni utente è inoltre caratterizzato dalle seguenti proprietà:

Followers per ogni utente U definiamo come $followers(U)$ l'insieme dei followers di U . Il loro numero è pari all'*in-degree* del nodo U .

Following similmente al precedente $following(U)$ indica l'insieme degli utenti seguiti da U . Il loro numero è pari all'*out-degree* del nodo U .

Direct Tag (dtag) $dtag(U, T)$ è l'insieme dei tweets che contengono un tag all'utente U generati al tempo T . Una descrizione più dettagliata è disponibile nella sezione riguardante i tweet [4.3].

Field Of View (FOV) $FOV(U, T)$ rappresenta l'insieme delle notizie visualizzate dall'utente U al tempo T , con $FOV \subseteq tweet(following(U), T_i) \cup retweet(following(U), T_j) \cup dtag(U, T_l)$ per $T_{i,j,l} \in \bar{T}$ lista di tempi non maggiori al tempo attuale T . Questo insieme è definito sia per una questione computazionale, ma anche per un motivo reale in quanto è difficile che un utente nell'arco della giornata riesca a vedere tutti i tweets e i retweets degli utenti che segue e i tweet che in cui risulta direttamente *taggato*. Questa affermazione diventa sempre più ragionevole al crescere della popolarità di un utente.

Ovviamente un grafo senza archi non rappresenterebbe nulla, così come un simulatore di social network senza interazione tra gli utenti; perciò di seguito viene mostrato come sono definiti gli archi e che caratteristiche hanno e più in seguito le possibili interazioni tra gli utenti.

Edges un arco tra due utenti U_1 e U_2 , scritto come (U_1, U_2) rappresenta la relazione di following tra il primo ed il secondo. Di conseguenza, ovviamente, risulta che $U_1 \in followers(U_2)$ e simmetricamente $U_2 \in following(U_1)$

Attachment ad un arco $e_j \in U \times U$ è associato un valore $attach(U_1, U_2) \in [0, 1]$ che rappresenta l'attaccamento di U_1 a U_2 , più questo valore si avvicina a 0 più la probabilità che U_1 smetta di seguire U_2 aumenta e viceversa. Quando l'arco viene creato il valore di *attach* è relativamente alto (si usa una distribuzione triangolare con parametri $a = 0.5$, $m = 0.8$, $b = 1$) in quanto ci si aspetta che un utente non smetta di seguire un altro utente poco dopo aver iniziato a seguirlo.

4.2 Attività di base

Elenchiamo qui le azioni che ogni utente può effettuare. Queste azioni vengono decise sulla base di estrazioni Monte Carlo: estratto un numero casuale, se ricade entro un certo intervallo di probabilità l'azione viene compiuta.

Tweet Posto $x = \frac{|followers(U)|}{|Users|}$ ogni utente U ad ogni tempo T ha la possibilità di creare un tweet secondo la seguente probabilità:

$$P_{tweet} = \alpha TZ(T) \left(-\frac{\log(x+4)}{\log(x+2)} + 2 \right) \quad (10)$$

Questo perchè ci si aspetta che un utente popolare sia più attivo di uno sconosciuto, per mantenere il suo livello di popolarità. Questa è una rappresentazione della realtà che contiene una restrizione, in quanto esistono esempi di persone poco conosciute che producono molti tweets, e di persone molto famose che non usano mai i social, soprattutto i più anziani. Tuttavia è ragionevole pensare che la precedente assunzione sia mediamente vera. La funzione 11 cerca di tradurre questo concetto in uno spazio di probabilità: più

il rapporto x tende a 1, più la funzione cresce, ma in maniera logaritmica, in modo che quando $x = 1$ si ha che $P_{tweet} \approx 0.5$; questo è per evitare la iperattività degli agenti, e prevenire così il completamento della rete. α è un fattore di normalizzazione, ottenuta sommando P_{tweet} a $P_{\neg tweet}$:

$$P_{\neg tweet} = \alpha(1 - TZ(T))(1 - (-\frac{\log(x+4)}{\log(x+2)} + 2)) \quad (11)$$

Retweet La probabilità $P_{retweet}$ è calcolata col medesimo meccanismo di P_{tweet} . Nel modello assumiamo per semplicità che un utente possa produrre al massimo un retweet ad istante di tempo. Una possibile versione futura potrebbe rilassare questo vincolo rendendo ancora più simile alla realtà il modello.

Tag Ogni tweet ha una possibilità di contenere un tag diretto ad un altro utente V. Tuttavia per questa caratteristica non è possibile fare delle assunzioni che possano essere verosimili, in quanto twetter permette ad un qualunque utente di taggare un qualunque altro utente. Perciò abbiamo deciso di implementare un sistema randomico dove ogni utente ha il 20 % di probabilità di taggare un altro qualsiasi utente. Grazie proprio a questa particolare caratteristica è possibile che accadano, sia nella realtà che nel modello casi particolari come quello descritto nella fase sperimentale.

4.3 Attività sociali

È importante definire quali sono le attività sociali che gli utenti possono fare per interagire tra di loro. Queste attività hanno come oggetto di scambio fra gli utenti i seguenti oggetti:

Tweet Il tweet dell'utente U al tempo T è definito come $tweet(U, T) = (j, likability, dislikability, dtag)$ dove:

- j è il *topic* su cui il tweet verte
- $likability \in [0, 1]$, indica la probabilità di quanto il tweet possa piacere agli utenti a cui interessa il *topic* j .
- $dislikability \in [0, 1]$, rappresenta la probabilità di quanto il tweet possa non piacere a chi non è interessato all'argomento
- $dtag$ indica l'utente V taggato nel tweet. Tale valore può anche essere nullo.

Retweet Il retweet effettuato dall'utente U al tempo T del tweet di V al tempo \bar{T} , definito come: $retweet(U, T) = tweet(V, \bar{T})$

Dtag Il tag diretto di un utente U ad un altro utente V al tempo T è definito come $dtag(U, V, T) = tweet(U, T)$ e rappresenta il caso in cui l'utente U ha “taggato” V con un $@U$. Questo permette all'utente V di vedere un tweet di U, anche nel caso in cui non lo seguisse.

In base agli oggetti definiti in precedenza possiamo definire le azioni di:

Post Nel momento in cui l'utente U è abilitato alla creazione di un tweet viene generato casualmente un *topic* j su cui verterà il tweet in modo tale che un topic di interesse per U sia selezionato con maggiore probabilità. Una volta scelto il topic j :

- se j è di interesse per U ($U.PI(j) \geq 0.5$) allora il *tweet* risultante avrà una *likability* elevata mentre la *dislikability* sarà casuale.

- viceversa il *tweet* generato avrà una *dislikability* alta ed una *likability* casuale.

Una volta definito il *tweet* viene generata la probabilità di avere un *dtag* ad un altro utente V. Se tale tag viene generato allora il *tweet* avrà un tag all'utente V, altrimenti il post non avrà alcun *dtag*.

Repost Per ogni utente U viene scelta casualmente una lista di k *tweet* $\bar{w} \in FOV(U)^k$, successivamente per ogni $\bar{w}_i \in \bar{w}$ viene valutata la possibilità di retweet di \bar{w}_i secondo l'equazione 11, in caso favorevole viene prodotto *retweet*(U, T) che sarà una lista di retweet effettuati da U al tempo T.

Unfollow Ad ogni istante T di tempo, ogni nodo ha lo 0.5% di possibilità di intraprendere un'attività di valutazione dell'attachment che intercorre tra lui e gli utenti da lui seguiti. La valutazione avviene secondo il seguente algoritmo:

1. Genera il *FOV* per l'utente U_1 .
2. Scorri tutti gli elementi (*tweet* o *retweet*) presenti nel *FOV*
3. Aggiorna l'attachment con il nodo U_2 che ha effettuato il *tweet* o il *retweet*.
 - (a) Se il topic dell'elemento è di interesse dell'utente (topic maggiore della media dei valori del *PI*) allora $attach(U_1, U_2)_{new} = \frac{attach(U_1, U_2)_{old} + likability}{2}$
 - (b) Se il topic dell'elemento no è di interesse dell'utente (topic minore della media dei valori del *PI*) allora $attach(U_1, U_2)_{new} = \frac{attach(U_1, U_2)_{old} + dislikability}{2}$
4. Elimina il collegamento con il nodo con il minore attachment

L'idea originale era quella di valutare l'attachment di tutti gli archi ad ogni istante T: questo è chiaramente un compito oneroso che, specialmente in reti con elevata densità, aumenta enormemente i tempi di esecuzione. Inoltre, non rispecchia appieno il processo di unfollow reale, poiché nessun utente umano valuta regolarmente l'interesse verso ogni suo collegamento. Per cui abbiamo abbandonato questa soluzione in favore dell'algoritmo appena descritto.

Follow Esistono due tipi di possibili modalità di following:

By Retweet (BR) Nel caso in cui nel *FOV* compaia un *retweet* di un utente non seguito, si può decidere di seguirlo attraverso la seguente valutazione: si considerano gli ultimi n *tweet* e *retweet* (a partire dall'istante $T = -20$) dell'utente target e da questi si inferiscono i suoi interessi. Dal momento che per un utente non è possibile conoscere il vettore di probabilità privato di un altro utente, esso viene inferito dal potenziale follower che ha intenzione di seguirne un altro valutando i topic dei suoi precedenti *tweet*, e la loro compatibilità viene valutata utilizzando la cosine similarity che, siccome ha a che fare con vettori di probabilità, assumerà valori nell'intervallo [0,1]. L'operazione di follow andrà a buon fine se il valore estratto è *minore* del valore di distanza calcolato in questo modo. Questa scelta è giustificata dal fatto che nella realtà non è possibile conoscere gli interessi di un'altra persona se non inferendola da ciò che condivide, oppure attraverso un fattore di intuito umano, che chiaramente gli utenti simulati non hanno.

Outside Factor (OF) avviene quando un utente U comincia a seguire un nodo V per fattori esterni al Social Network, quali ad esempio nuova amicizia nella vita reale, nuovo follow su altri mezzi di comunicazioni online, ecc. Questa operazione è speculare all'unfollow: ad ogni istante di tempo T ogni nodo ha lo 0.5% di possibilità di iniziare a seguirne un altro, estratto a caso dall'insieme di utenti.

4.4 Step Simulazione

Usando tutte le azioni definite in precedenza possiamo ora descrivere gli step che avvengono nella simulazione ad ogni istante di tempo T.

Tweet Step (TS) Per ogni utente U viene generato una probabilità casuale $P_t(U, T)$ che rappresenta la sua inclinazione di produrre un *tweet* al tempo T. Se $P_t(U, T)$ risulta minore alla probabilità P_{tweet} (eq. 11) allora viene generato il $tweet(U, T)$ come descritto in **Post**.

Retweet Step (RS) Similmente alla fase precedente, per ogni utente U viene generato una probabilità casuale $P_r(U, T)$ che rappresenta la sua inclinazione di produrre un *retweet* al tempo T. Se $P_r(U, T)$ risulta minore alla probabilità $P_{retweet}$ (eq. 11) allora viene generato il $retweet(U, T)$ come descritto in **Repost**. In questa fase avvengono anche le eventuali valutazioni del **Follow by retweet**, con la creazione di nuovi archi.

Evaluation Step (ES) Questa fase comprende le altre due operazioni descritte, ovvero **Follow by Outside Factor** e l'**Unfollow**.

Le azioni descritte in precedenza devono necessariamente essere eseguite sequenzialmente una dopo l'altra, tuttavia i singoli step possono essere facilmente parallelizzati, senza problemi di concorrenza, in quanto il post di un tweet è indipendente per ogni utente; lo stesso vale per il retweet e infine l'aggiornamento dell' *attach* è dipendente dai tweet e i retweet effettuati **TS** e **RS**, ma i singoli aggiornamenti sono indipendenti tra gli utenti. Per questo motivo è possibile parallelizzare il modello in modo che le azioni contenute in ogni step di simulazione vengano effettuate contemporaneamente, ma comunque rispettando l'ordine descritto. Questa modello di parallelizzazione non è tuttavia implementato per questa versione in quanto sarebbe necessario implementare tecniche di sincronizzazione tra le strutture dati che contengono gli elementi della simulazione.

5 Interpretazione multi-agente

Prima di descrivere gli agenti è necessario caratterizzare l'ambiente in cui essi operano. In [24] vengono descritti i parametri con cui un agent environment si caratterizza. Nel modello descritto in precedenza l'ambiente è formalizzato come un grafo diretto in cui i nodi sono gli agenti e gli archi sono le connessioni tra di essi e il peso è la forza del loro legame. Questo tipo di ambiente è tendenzialmente inaccessibile in quanto gli agenti hanno visione solo dell'ambiente che li circonda; però ci sono delle eccezioni. È possibile che un utente abbia accesso ad una parte di ambiente a lui distante nel caso in cui venga taggato da una persona con cui non ha nessun legame, questo è il principio su cui la fase 2 degli esperimenti si basa. Esiste inoltre il caso degenero in cui un particolare agente è connesso a tutti i nodi e quindi ha potenzialmente accesso a tutto l'ambiente, tuttavia in questo caso la sua visione è limitata dal suo FOV. Per via del fatto che gli utenti basano le proprie azioni su un modello probabilistico l'ambiente risulta non deterministico, in quanto non è possibile prevedere con certezza il suo stato successivo. Possiamo definire l'ambiente come episodico in quanto ogni utente prende decisioni in base allo stato in cui attualmente si trova la simulazione senza considerare le i possibili effetti nel futuro. Risulta complicato definire se gli agenti si trovino in contesto statico o dinamico o dinamico e discreto o continuo, in quanto non si applica completamente a nessuna delle due possibili definizioni. L'ambiente risulta statico perché non ci sono processi che sono al di fuori del controllo degli utenti (tale concetto non è più veritiero nel caso in cui venga implementata in una versione futura la crescita del grafo iniziale), tuttavia gli agenti producono degli artefatti

che modificano in qualche modo l'ambiente e le future interazioni delle sue componenti. Inoltre possiamo definire la simulazione come discreta, in quanto esiste un numero predefinito di azioni che gli utenti possono compiere, ma le percezioni che gli utenti hanno dello stesso cambiano ad ogni step. I metodi di interazioni permesse sono descritte più in dettaglio nel seguito.

Gli agenti all'interno del modello ABSoNeS possono essere visti come agenti un ibrido tra hysteretic e knowledge-based [8]. Le loro azioni nel modello agent-based vengono classificate come:

See Gli agenti, o utenti in questo caso, osservano l'ambiente che li circonda; ovvero guardano chi li segue e chi loro seguono. Inoltre vedono gli elementi presenti nell'ambiente quali tweet, retweet e tag diretti. Tuttavia hanno una visione limitata di ciò che li circonda. Sono infatti in grado di visionare solo l'ambiente che li circonda che in questo caso corrisponde agli altri utenti ai quali sono connessi e alle loro azioni.

Action Le modalità in cui gli agenti compiono le loro azioni sociali sono il motivo per cui possono essere classificati come ibridi. Questo è dovuto al fatto che gli utenti decidono che tweet produrre in base a loro stati interni: activity e personal interest. Tali caratteristiche influiscono sulle azioni personali che ogni utente compie e solo la parte isteretica dell'agente. Quando però un agente si trova nella sua fase evaluation prende decisioni in base ad una conoscenza che inferisce da ciò che lo circonda. In particolare la decisione di continuare a seguire o di smettere di seguire qualcuno è dovuta da un ragionamento effettuato sui tweet, prodotti da altri agenti, che osserva. E il suo valore di attachment varia a seconda di un ragionamento, per quanto banale, basato di essi. La parte più caratterizzante deriva dal ragionamento che ogni agente effettua quando decide di seguire qualcuno. Infatti guardando le azioni di un altro utente inferisce quali possano essere i suoi interessi e se sono compatibili a propri. In caso positivo può decidere di seguirlo, altrimenti no.

Do Le azioni che ogni utente può effettuare sono largamente descritte nella sezione di modellazione, tuttavia riassumendo ogni utente è in grado di seguire, smettere di seguire; produrre un tweet o un retweet; taggare direttamente un altro utente.

Internal Alcuni stati interni dell'utente sono immutabili, ovvero i personal intereset e le time zone. Tuttavia dopo ogni step di simulazione ogni agente aggiorna i suoi valori di attachment verso gli altri utenti.

Gli agenti nel modello sono da considerarsi autonomi su alcuni dei contesti che li caratterizzano; in particolare ogni agente è in grado di decidere le proprie azioni in base ai suoi stati interni e alla visione dell'ambiente che lo circonda; ha inoltre controllo su alcuni dei suoi stati interni, anche se questo controllo è dovuto ad una risposta all'ambiente e non ad una sua decisione. Tuttavia gli utenti non hanno la capacità di prendere decisioni in base alla loro esperienza regressa, ma in ogni caso questo fattore non è particolarmente rilevante in quanto possiamo facilmente osservare che spesso nemmeno gli utenti reali dei social network imparano dalle loro azioni. Basti pensare al presidente Trump, che nonostante abbia usato per anni il suo account tweeter per auto-diffamarsi, presumibilmente senza rendersene conto, continua a fare lo stesso nonostante l'importanza della carica che copre.

Per completezza di descrizione risulta essenziale osservare che gli agenti non hanno un obiettivo, né in comune né personale. Potrebbe sembrare che ogni utente abbia come obiettivo quello di aumentare sempre di più la sua cerchia di followers anche se in realtà, per come è definito il modello, gli agenti non hanno nessun interesse verso questo obiettivo e nemmeno sono incentivati ad averlo. Così come per la maggior parte degli utenti reali, gli agenti sono solo interessati ad esprire le loro opinioni senza un obiettivo particolare. Per lo stesso motivo le

azioni che compiono non hanno una utilità nei confronti di sé stessi, dell’ambiente e degli altri utenti, semplicemente tweetano per il solo piacere di farlo.

È difficile caratterizzare in termini classici l’interazione tra gli utenti, in quanto essi non hanno interazione diretta e non si passano nessuna informazione tra loro. Si potrebbe dire che gli agenti comunicano attraverso degli artefatti condivisi (tweet, ecc) avvicinandosi al modello artifact-mediated indirect interaction; tuttavia, nonostante gli artefatti siano condivisi e ci sono delle chiare regole che ne definiscono l’accesso da parte degli agenti, essi non sono un vero e proprio canale di comunicazione ma risultano essere delle “tracce” a cui altri agenti decidono se prestare attenzione o meno. Questo è dovuto dal fatto che non sia presente nessun obiettivo, comune o personale a cui tutti fanno riferimento.

6 Esperimenti

In questa sezione esponiamo le prove condotte con il nostro modello. Abbiamo organizzato gli esperimenti in due fasi:

Fase 1 Nella prima fase testiamo la solidità del nostro approccio andando a verificare che la rete rispetti le caratteristiche tipiche di una rete sociale.

Fase 2 Nella seconda fase osserviamo il cambiamento del sistema all’occorrenza di un particolare evento, in particolare studiamo il “salto” di popolarità da parte di un utente.

Per quanto riguarda la prima fase, uno dei rischi maggiori è produrre un modello incline al *completamento della rete*, ovvero un modello i cui agenti, alla fine della simulazione, sono tutti collegati fra di loro da due archi (essendo un grafo diretto). In un social network reale questa cosa non si verifica, in quanto gli utenti non hanno interesse ad essere collegati con tutti gli altri. Questo è stato uno dei principali obiettivi durante la costruzione del modello, assieme alla conservazione della proprietà di *invarianza di scala* e alla tendenza degli utenti a stabilire collegamenti secondo l’*omofilia*.

Nella seconda fase simuliamo lo scenario in cui un utente con pochi collegamenti in entrata, ovvero con pochi *follower*, viene “notato” da uno degli “hub”, ovvero un utente con molti follower. Per far ciò, forziamo il retweet da parte dell’utente hub di un tweet creato ad-hoc per l’utente meno importante; dopodichè controlliamo la variazione di popolarità dell’utente prescelto in termini di follower. Questo per simulare quanto avvenuto nel 2012 tra l’On. Maurizio Gasparri e l’utente Daniele Termite [22] dopo essere stato “ripreso” su Twitter dall’Onorevole, l’utente Termite ha visto crescere improvvisamente di popolarità il suo account, con un boom di follower nel giro di brevissimo tempo, per poi assestarsi su un numero di circa 10.000 seguaci. La replica *in-silico* di questo fenomeno utilizzando il nostro modello può essere considerata come indica di affidabilità dello stesso.

Per la realizzazione del modello e la conduzione degli esperimenti la scelta implementativa è ricaduta su *Python* (versione 3) e sulla libreria *networkx* [19], un package molto curato e completo per la modellazione di reti. In principio avevamo optato per l’ambiente di simulazione PyCx, che fornisce un’interfaccia per la conduzione di esperimenti e la visualizzazione dei risultati, ma abbiamo riscontrato successivamente una incidenza negativa sulle performances del modello; per cui abbiamo virato su una CLI per l’inserimento dei parametri. La visualizzazione dei grafici è stata effettuata con il software Gephi, mentre i grafici delle metriche con R.

6.1 Fase 1

Per valutare la buona riuscita degli esperimenti, considerando quanto prodotto dalla letteratura, abbiamo individuato come parametri principali per valutare l'attendibilità della rete prodotta dal nostro modello le seguenti metriche:

1. La *densità* della rete, per monitorare la tendenza del modello al completamento o meno della rete.
2. I parametri della *power law* che intercorre sul grado dei nodi, per studiarne l'invarianza di scala.

Altre metriche che abbiamo incluso nell'analisi sono:

1. Il *coefficiente di clustering*, per valutare la predisposizione alla formazione di gruppi di agenti.
2. L'*assortatività per attributo*, per valutare l'omofilia rispetto ai topic caratterizzanti gli utenti. Come attributo abbiamo scelto il topic con valore più alto nel *PI* di ogni nodo.
3. L'*assortatività per grado r(out,in)*, a supporto del punto 2 della lista precedente, ovvero la valutazione dell'invarianza di scala.
4. Il *cammino minimo medio* (in caso di reti connesse) per verificare l'ipotesi di *small world*

Abbiamo quindi organizzato una sessione di esperimenti con i seguenti parametri comuni:

- 1000 utenti
- 10 topic per utente
- 1080 iterazioni (equivalenti a 90 giorni)

La differenza tra le varie run sta nella diversa istanziazione della rete iniziale; non avendo indicazioni sul valore di densità del grafo reale di Twitter, abbiamo optato per due soluzioni principali:

- Istanziazione di una rete scale-free, a densità fissa
- Istanziazione di una rete random, a densità variabile

Il package networkx di Python permette l'istanziazione di una rete scale-free con una densità non modificabile pari a circa 0.1%: questo è dovuto al particolare modello usato dalla libreria, descritto in [23], che per rispetto alla power law non permette la scelta del numero di archi iniziale. A questa abbiamo quindi affiancato una serie di prove con una random network come configurazione di partenza, avente valori diversi come densità iniziale: nello specifico, i valori sono 0.1%, 0.5%, 1.0% e 5.0%.

6.1.1 Risultati

Le figure 3, 4, 5, 6 e 7 mostrano l'andamento dei quattro indici che abbiamo scelto per la valutazione. Nell'esperimento con istanziazione della rete iniziale a invarianza di scala (figura 3) si può notare come la densità cresca in modo piuttosto contenuto sebbene in modo debolmente esponenziale: da una densità iniziale di circa 0.1% si arriva a circa 0.5%. Il coefficiente di clustering invece ha un andamento particolare: decresce fino a poco prima della metà del periodo considerato, per poi iniziare una crescita irregolare fino alla fine dell'esperimento. Una

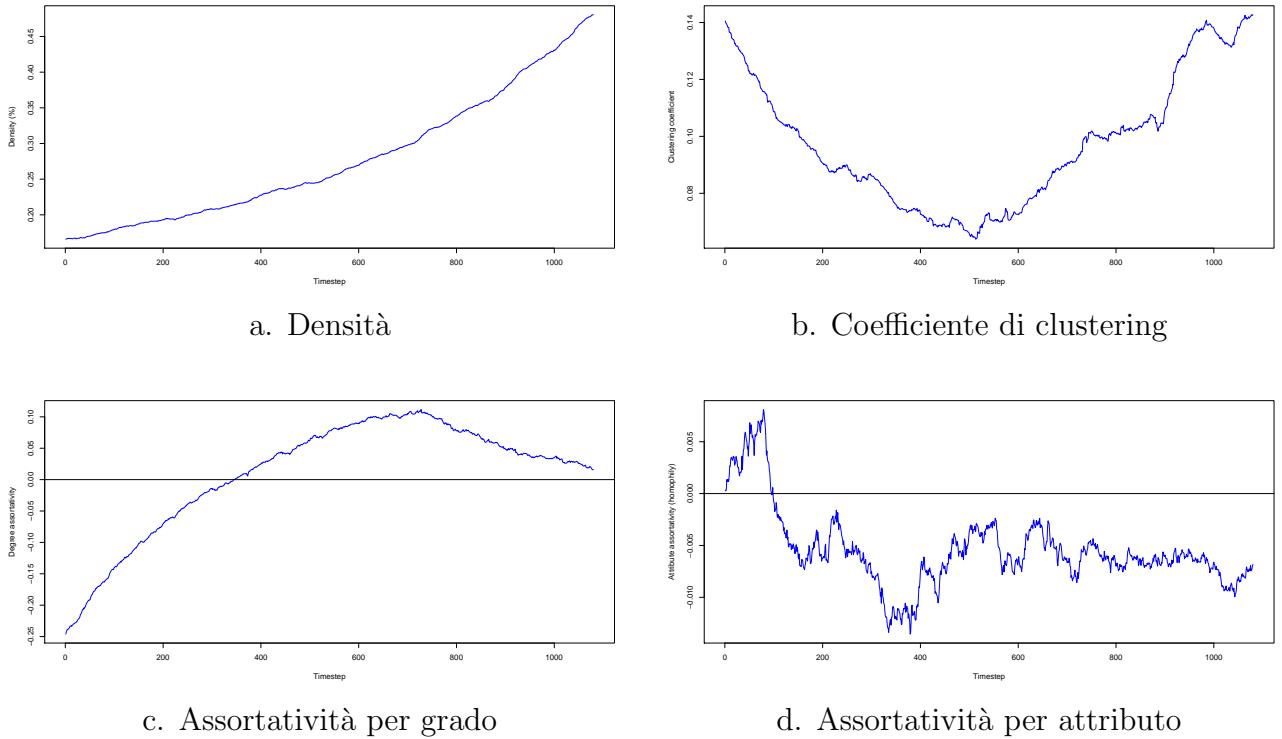


Figura 3: Indici scale-free network, densità 0.1%

spiegazione di questo fenomeno potrebbe essere il fatto che, in principio, la rete abbia molti nodi senza alcun collegamento in entrata o in uscita; inoltre, una porzione ancora maggiore di questi nodi ha solo pochi collegamenti in uscita verso i due, tre nodi “hub” che fanno capo alla maggior parte degli archi. È plausibile quindi che, fino ad un certo punto, i collegamenti non vadano a rafforzare comunità già presenti, quanto a unire per la prima volta nodi distanti. Al contrario, l’assortatività per grado cresce regolarmente fino all’iterazione 750 circa, per poi cominciare a calare leggermente: questo è indice del fatto che a partire da quel momento, l’in-degree medio dei nodi con molti collegamenti cresce in modo più rapido dell’out-degree dei nodi con pochi collegamenti, con effetto negativo sulla misura di assortatività $r(out,in)$. Questo dovrebbe essere indicatore di una tendenza della rete a conservare una configurazione ad invarianza di scala.

A differenza degli altri indici, l’assortatività per attributo è molto irregolare: dopo un brusco calo che la porta ad assumere valori negativi, continua ad oscillare in prossimità dello zero, senza mai tornare a valori positivi. Questo è segno che i nodi tendono a formare legami a prescindere dal topic più rappresentativo di ognuno.

L’esperimento successivo (figura 4) condivide con il primo il valore di densità iniziale, mentre differisce per costruzione: è infatti una random network priva, quindi, di nodi “hub”; si riscontrano comunque nodi con più collegamenti in entrata e in uscita rispetto agli altri, ma la differenza, in questi termini, è molto ridotta. Ciò si rispecchia in una crescita molto più contenuta e lineare della densità (si arriva all’ultima iterazione con circa lo 0.3%), con i nodi che, avendo tutti un *field of view* limitato, fanno fatica a stabilire nuovi collegamenti con nodi affini dal punto di vista degli interessi. Da qui si può trarre la conclusione che la presenza di nodi “popolari” faciliti l’instaurazione di nuovi legami tra nodi meno importanti. Inoltre, nell’andamento del coefficiente di clustering, si riscontra lo stesso fenomeno osservato in precedenza sulla scale-free network: dopo un calo iniziale, l’indice torna a crescere, ma in modo

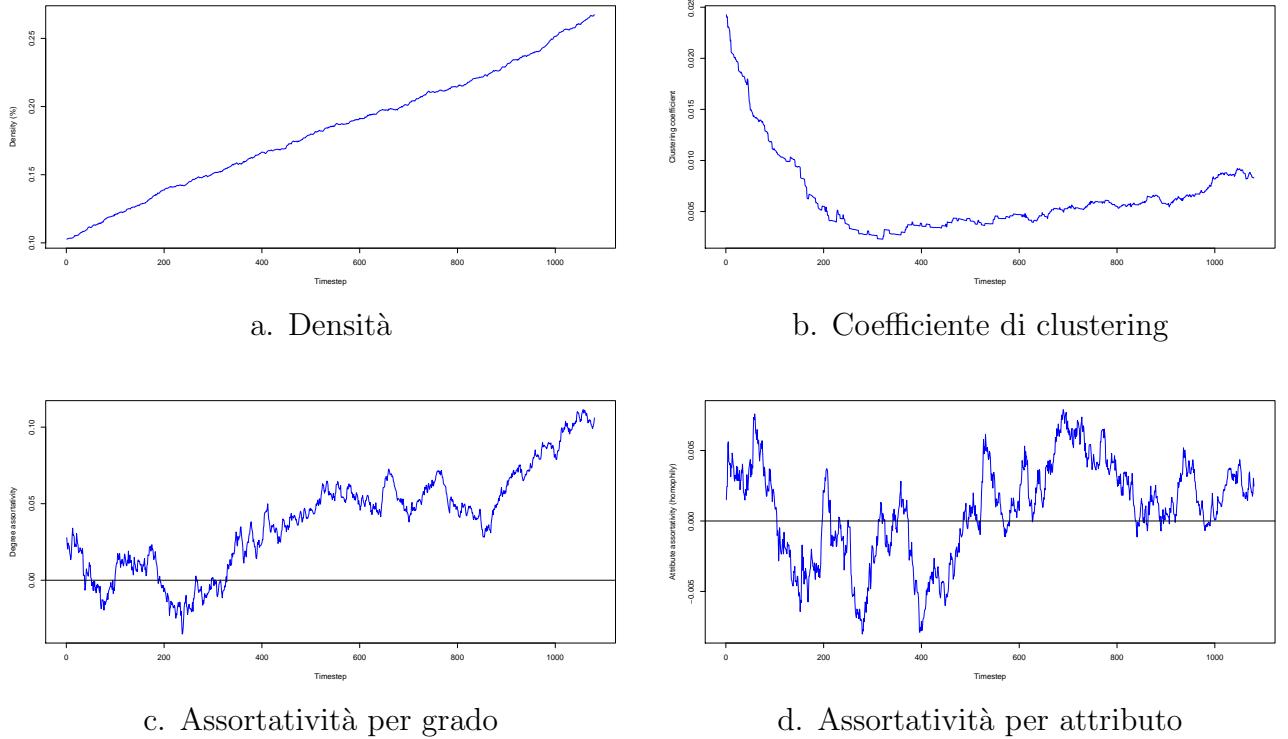


Figura 4: Indici random network, densità 0.1%

sensibilmente più lento rispetto all'altro caso. Da notare anche il valore di partenza, circa cinque volte minore. Anche questo comportamento può essere imputato alla scarsa densità e all'elevata sparsità degli archi, per cui le comunità vengono costituite con fatica. L'assortatività per grado mostra un andamento incerto ma crescente, a indicare che i nodi tendono a legarsi in base alla similarità tra out e in degree. L'assortatività per attributo è più incerta rispetto al caso precedente, con la differenza che nella seconda metà della simulazione rimane quasi sempre positiva.

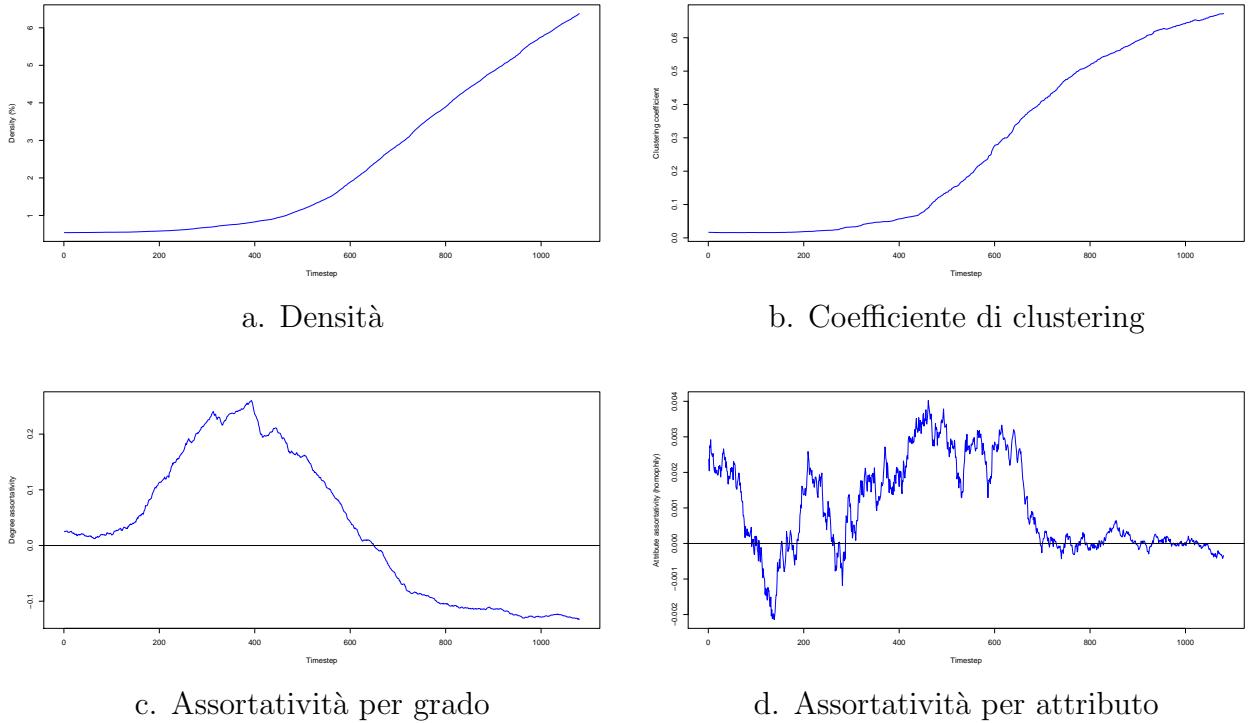


Figura 5: Indici random network, densità 0.5%

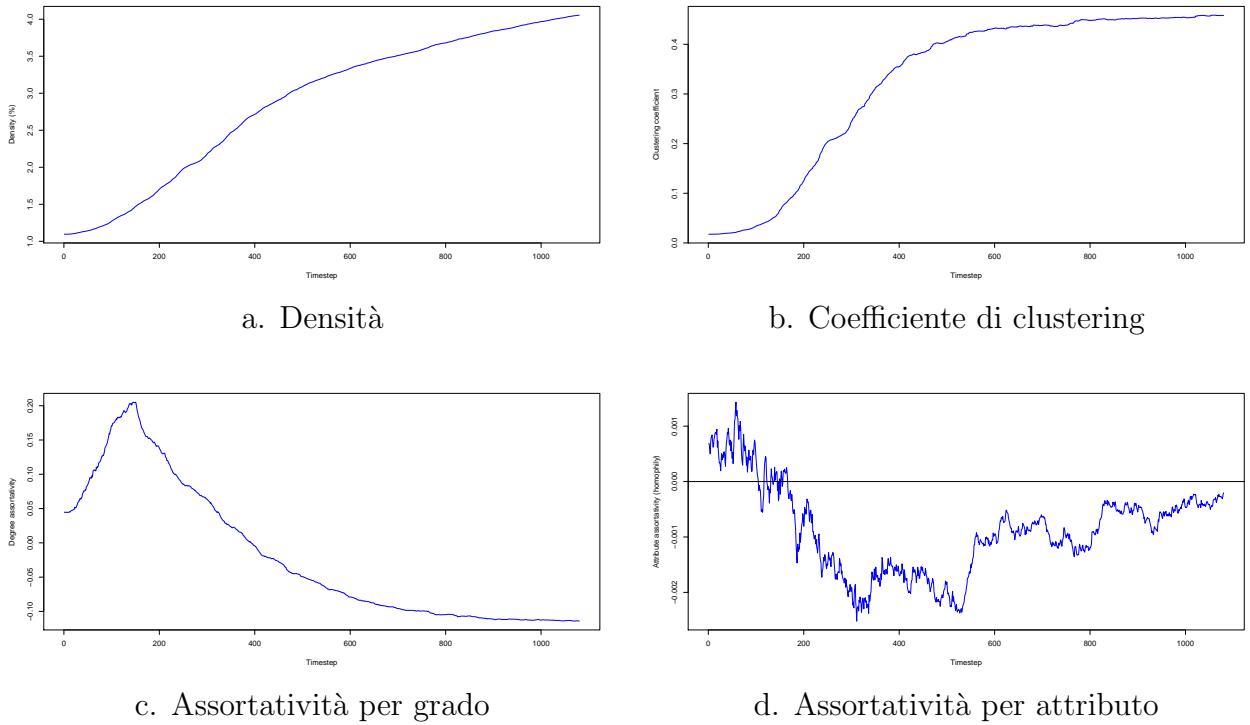


Figura 6: Indici random network, densità 1.0%

Il caso con densità di partenza pari a 0.5% (figura 5) è piuttosto interessante, in quanto vi abbiamo osservato la crescita più rapida per quanto riguarda la densità: dallo 0.5% iniziale, dopo una fase relativamente calma, cresce fino a superare il 6% alla fine della simulazione. Questo si riflette sugli altri indici: ad esempio, il coefficiente di clustering, dopo un rapido incremento con una pendenza pari a quella assunta dalla densità, comincia a ridurre l'ascesa.

Questo suggerisce che, man mano che la densità si fa importante, i collegamenti avvengono sempre di più all'interno di comunità già formate. Non solo: a partire dall'iterazione 984 la rete ha anche raggiunto lo stato di completa *connessione*, ovvero la rete è diventata un'unica componente connessa; questo concorre all'affievolimento della crescita del coefficiente di clustering, in quanto favorisce l'instaurazione di archi intra-comunità. Nell'assortatività per grado si ritrova il fenomeno osservato nella prima run: l'indice cresce fino a raggiungere un picco, per poi calare fino alla fine delle iterazioni, a indicare una tendenza della rete ad assumere una struttura ad invarianza di scala. Qui però il fenomeno è più rapido: il picco è più alto e viene raggiunto prima, mentre nella fase calante si verifica un livellamento improvviso, riscontrato anche nella misura di omofilia, dove l'assortatività per attributo si stabilizza sullo 0; questo comportamento può essere imputato al fatto che la rete ha raggiunto uno stato in cui le differenze tra i nodi, sia dal punto di vista della metrica $r(out, in)$ che dell'assortatività per attributo, sono meno accentuate; in altre parole, l'invarianza di scala comincia ad indebolirsi. Inoltre, la distanza tra i nodi diventa molto piccola, per cui è più facile per un agente scovare un altro agente simile per topic e instaurare un collegamento.

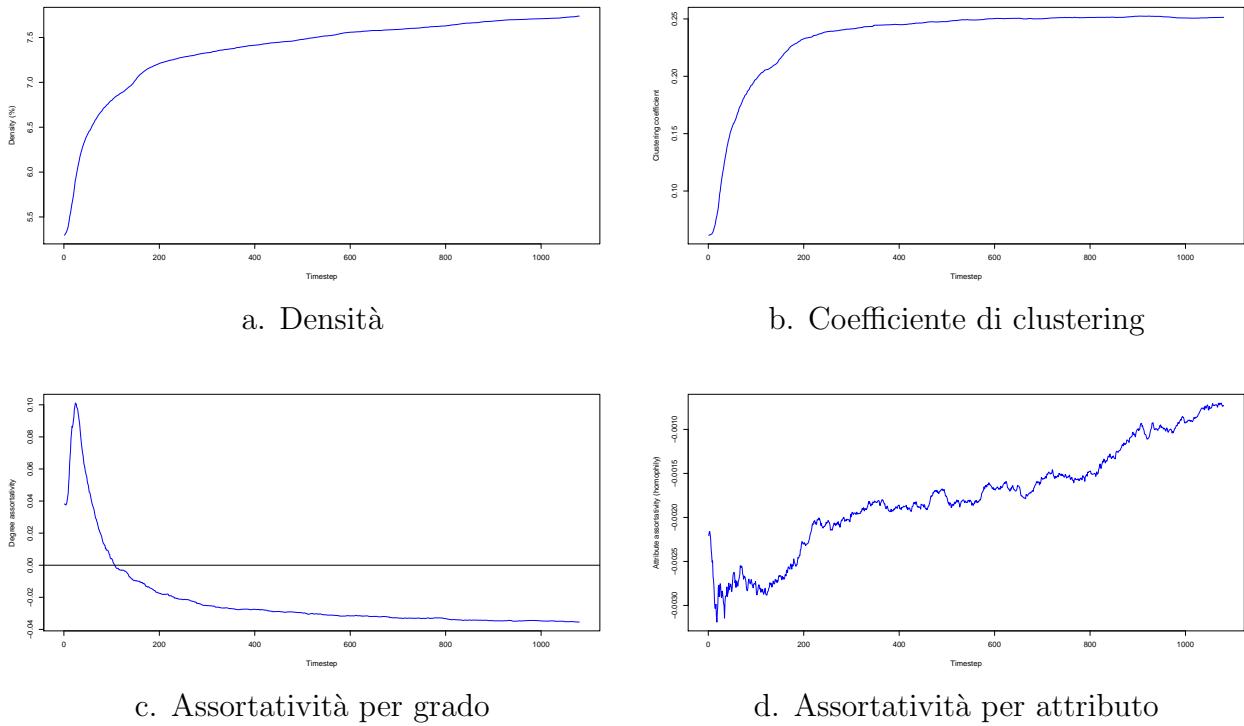


Figura 7: Indici random network, densità 5.0%

Le run con densità di partenza pari all'1.0% e al 5.0% (figure 6 e 7) hanno prodotto risultati piuttosto simili, e condividono un fenomeno importante: la densità, dopo un'iniziale fase di crescita, tende a rallentare la sua corsa con approssimazione logaritmica, attestandosi, nell'ultimo caso, poco sotto all'8.0%. Sembra, quindi, che ci sia un limite alla crescita del numero di archi, per cui il completamento della rete (densità 100%) sembra essere scongiurato. Il rallentamento viene percepito anche dal coefficiente di clustering, che decelera vistosamente fino a livellarsi. L'assortatività di scala presenta lo stesso fenomeno del primo e del terzo esperimento, ma lo sviluppo è più repentino man mano che si cresce con la densità iniziale: più collegamenti permettono ai nodi di avere più occasioni di instaurarne altri, per cui il processo di trasformazione da random network a rete ad invarianza di scala è più rapido. L'assortatività per attributo è

poco indicativa, in quanto in un caso descresce e da positiva diventa negativa, per poi risalire in modo incerto, mentre nell'altro cresce costantemente ma resta negativa.

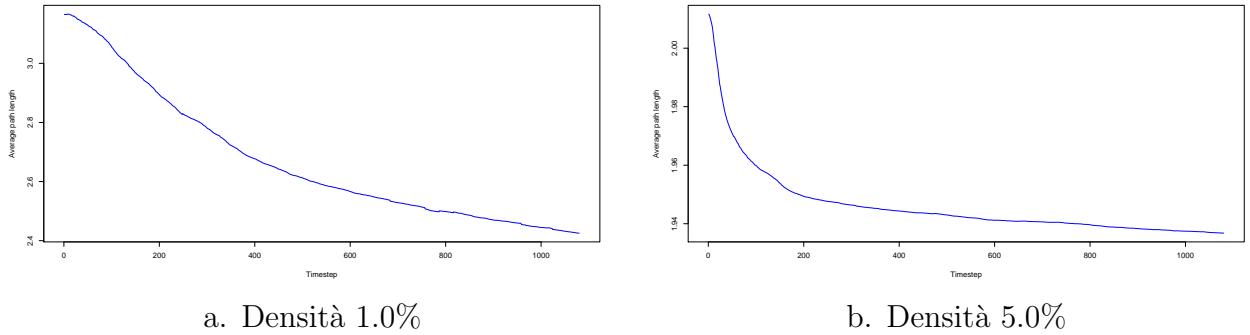


Figura 8: Cammino minimo medio

Per questi ultimi due esperimenti è stato possibile calcolare fin dalla prima iterazione il *cammino minimo medio* (figura 8). Com'era prevedibile, in entrambi i casi è sempre decrescente, per via della crescente densità; è interessante notare come nel caso con densità 5.0% il cammino minimo medio scenda al di sotto della soglia dei due step. Esiste perciò un numero non indifferente di cammini composti solo da un arco, per cui ogni nodo ha accesso nel suo *fov* una grande quantità di retweet provenienti da nodi inizialmente piuttosto lontani. Ci si aspetterebbe quindi che questo porti a una crescita esponenziale delle connessioni, in realtà le limitazioni poste nel modello riescono a limitarne l'esplosione.

Richiamando quanto detto nella sezione 3.2, una rete è detta ad invarianza di scala, o scale-free, se la distribuzione dei gradi (in-degree o out-degree) dei suoi nodi segue una power law. Le figure 9, 10, 11, 12 e 13 mostrano il plot della distribuzione iniziale (verde) e della distribuzione finale (rossa), dell'*in-degree* dei nodi in ogni esperimento di questa sezione, normalizzati per il numero massimo di in-degree in modo da poterne confrontare l'andamento. È importante notare che la curva verde della figura è stata generata secondo una power law, quindi rappresenta un modello di rete ad invarianza di scala “ideale”; normalizzata in questa maniera, può quindi essere presa come modello di riferimento anche per gli altri esperimenti, perciò l'abbiamo riportata (in blu) in tutti gli altri grafici.

La prima esecuzione del modello (figura 9) non va a modificare sostanzialmente la natura scale free della rete, in quanto la distribuzione in uscita è solo leggermente deformata. Nella seconda run (figura 10) è lampante come la costruzione random iniziale sia distante dal modello a invarianza di scala, e, nonostante l'esecuzione del nostro modello, la distribuzione dell'in-degree non cambia. Nella terza run (figura 11) si ha una situazione di partenza analoga alla precedente, con la sola differenza della densità; dopo l'esecuzione del modello però, la curva è molto più simile ad una power law rispetto al caso precedente. Il nostro modello sembra quindi adattare, con un aiuto derivante dalla relativamente elevata densità, una random network ad una rete scale-free. Questo fenomeno viene ulteriormente accentuato dal caso con densità di partenza all'1.0% (figura 12): nonostante una curva di partenza di tipo randomico, la curva finale si discosta di poco da quella ideale. Un discorso a sè stante spetta per l'ultima run (figura 13), dove la rete randomica istanziata in partenza presenta già una distribuzione dell'in-degree molto simile ad una power law, e l'esecuzione del nostro modello non fa altro che accentuare la distanza tra i nodi “popolari” e quelli meno importanti.

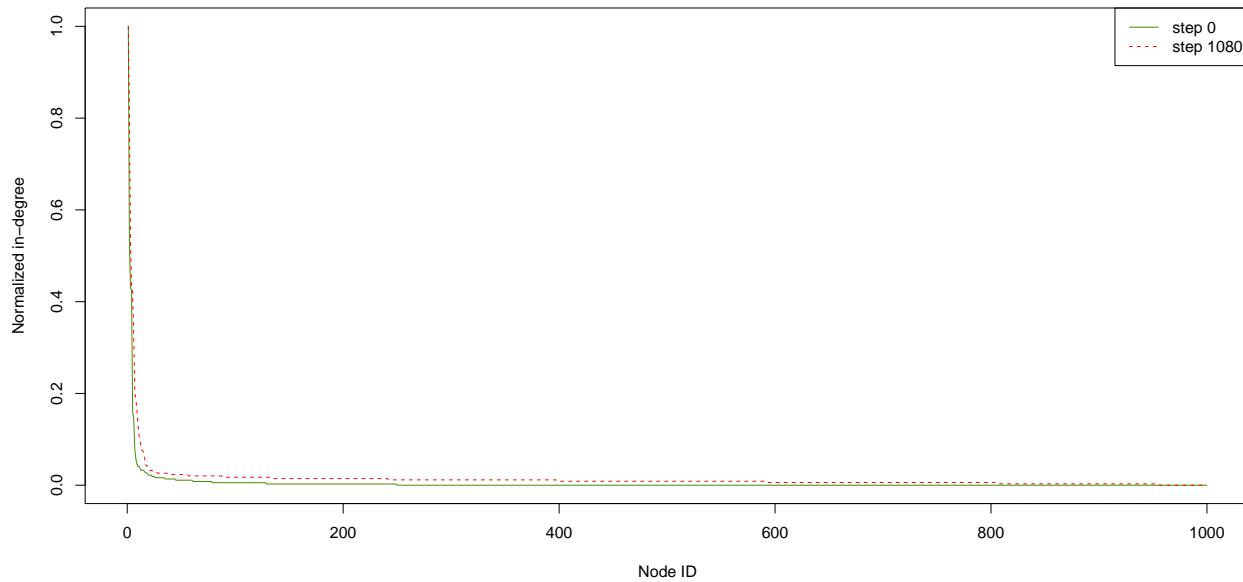


Figura 9: Distribuzione in-degree della rete scale-free, densità 0.1%

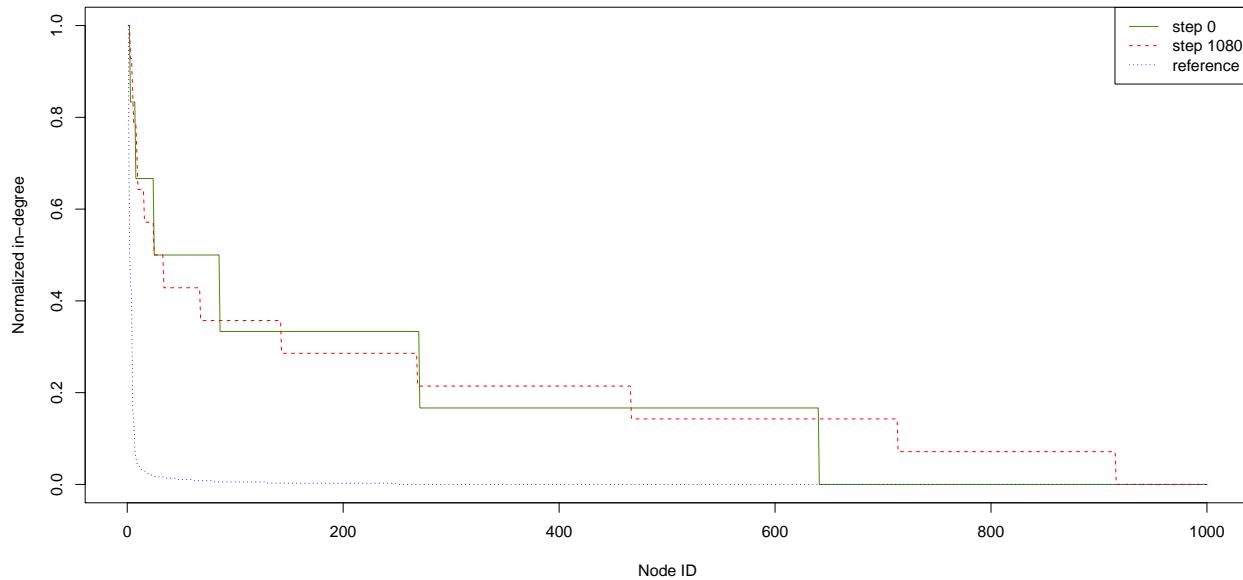


Figura 10: Distribuzione in-degree della rete random, densità 0.1%

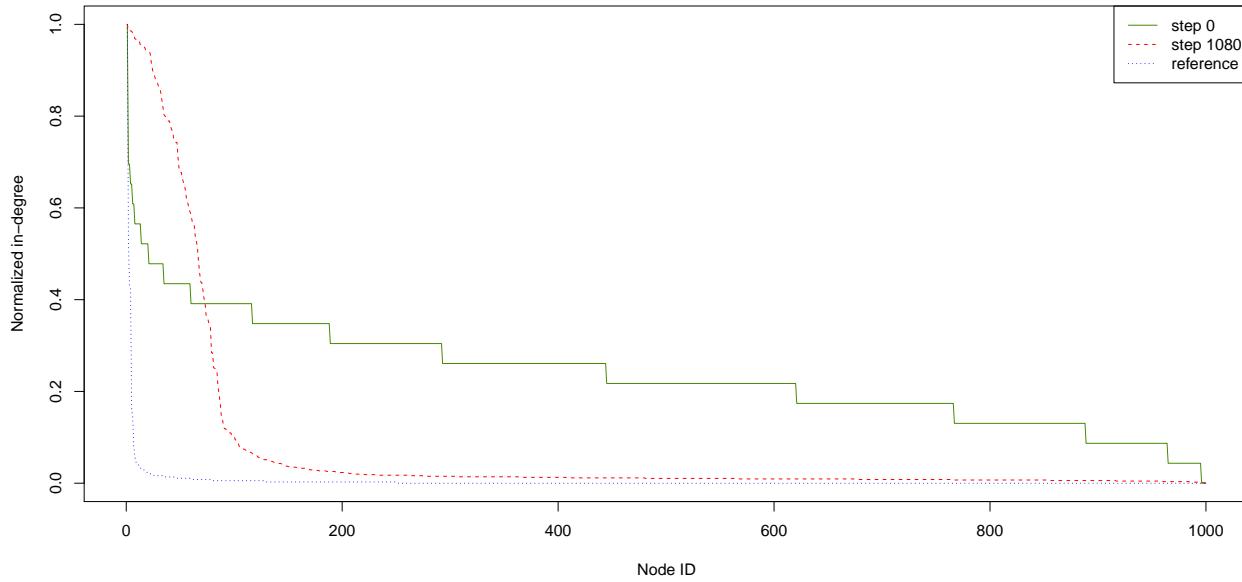


Figura 11: Distribuzione in-degree della rete random, densità 0.5%

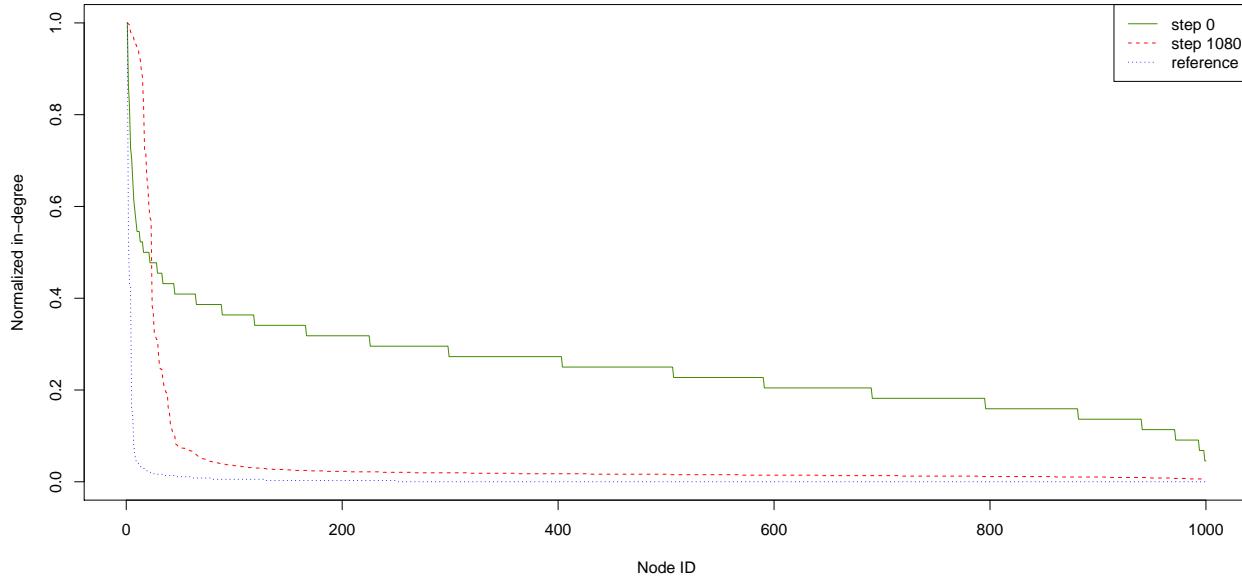


Figura 12: Distribuzione in-degree della rete random, densità 1.0%

Sembra quindi che ci sia una certa correlazione tra l'indice di assortatività per grado visto in precedenza e l'andamento delle distribuzioni dell'in-degree, in quanto i casi che presentavano un comportamento “a cuspide” nella misura di assortatività tendono a rispettare il comportamento di una rete ad invarianza di scala, in particolare la run con rete scale-free e le run con rete random con valori di densità 0.5%, 1.0% e 5.0%.

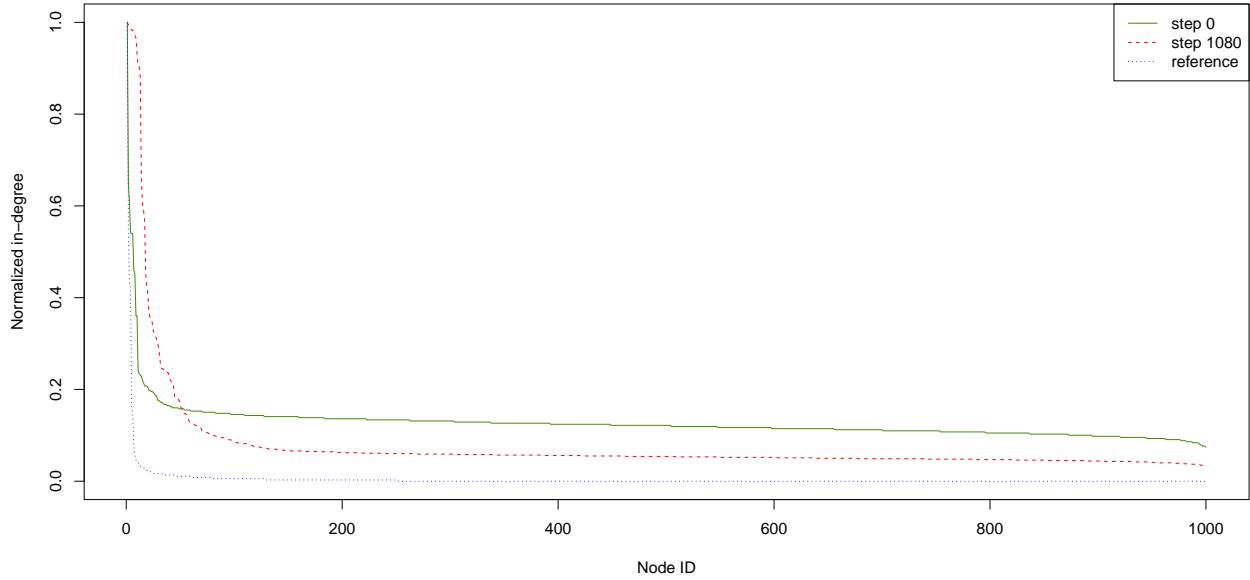


Figura 13: Distribuzione in-degree della rete random, densità 5.0%

Nella sezione 3.2 abbiamo esposto la definizione di invarianza di scala e di power law. Dai dati prodotti dagli esperimenti è possibile stimare i valori C e γ delle curve finali (in rosso) rappresentate nei plot precedenti. Nella libreria *scipy* di Python è presente un’implementazione del *metodo non lineare dei minimi quadrati*: l’abbiamo utilizzata per stimare i parametri delle curve. Nella tabella sottostante riportiamo i valori dei coefficienti C e γ della power law per le distribuzioni finali (in rosso) di tutte le run effettuate *con instanziazione randomica della rete*

Densità	0.1%	0.5%	1.0%	5.0%
C	22.11	1981.45	1832.28	1722.57
γ	0.36	0.55	0.63	0.55

In quest’altra tabella riportiamo i valori per le distribuzioni iniziale e finale della run effettuata *con instanziazione scale-free della rete*:

Step	0	1080
C	386.26	390.65
γ	1.09	0.87

Da notare che la prima colonna di questa tabella contiene i parametri della curva di riferimento. Per quanto riguarda le reti random, quella che alla fine delle iterazioni rispetta più fedelmente l’invarianza di scala è quella con densità di partenza posta all’1.0% (figura 14). La migliore in assoluto è invece la run con la rete scale-free iniziale (figura 15). Dai plot salta immediatamente all’occhio la differenza di densità tra le due soluzioni, e il diverso numero di nodi “hub”.

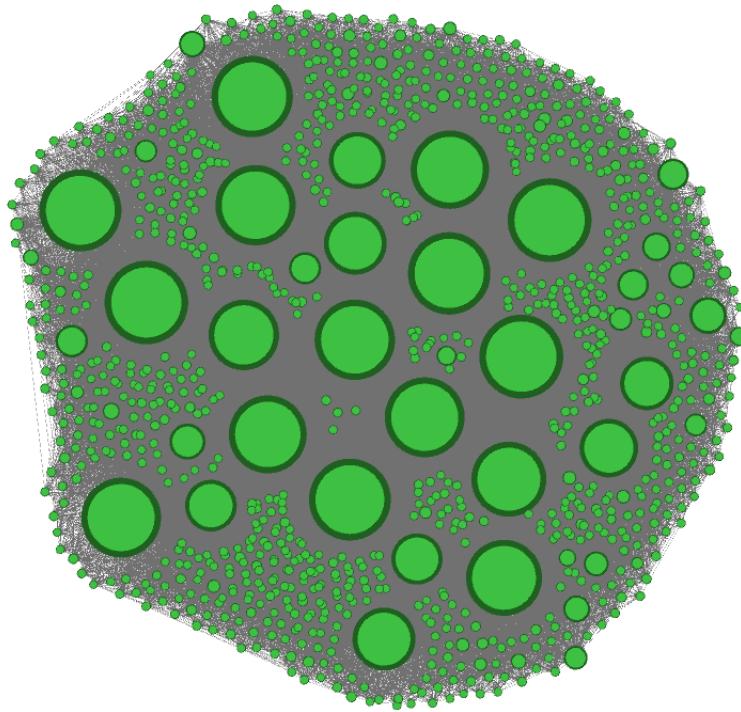


Figura 14: Grafo rete random, densità 1.0%

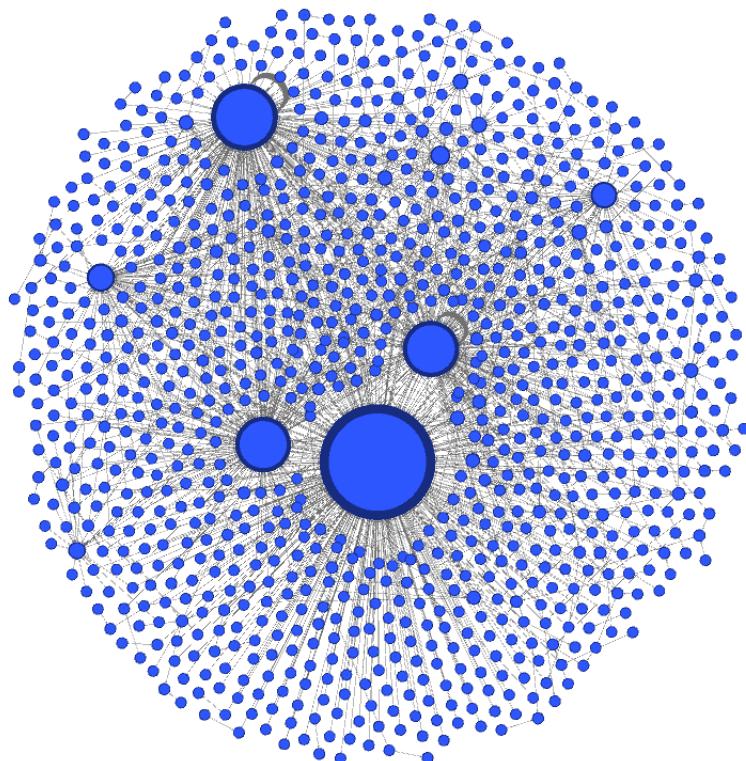


Figura 15: Grafo rete scale-free, densità 0.1%

6.1.2 Commento

Da questi esperimenti possiamo trarre le seguenti considerazioni sugli obiettivi che ci eravamo posti:

Completamento della rete Il nostro modello riesce a porre un limite alla crescita del numero di collegamenti; la rete, e quindi gli agenti, raggiungono una certa stabilità in prossimità di alcuno dei valori di densità (circa tra il 7% e l'8%). Non avendo idea dell'entità della densità del grafo reale di Twitter, si può solo supporre che un certo valore di densità sia quello esatto; probabilmente un valore tra il 7% e l'8% è ancora pittosto alto, c'è però margine per migliorarlo, andando a restringere ancora di più i parametri descritti nella sezione 4.

Invarianza di scala Nella prima run, il nostro modello ha dimostrato che, data una configurazione di partenza del grafo a invarianza di scala, riesce a mantenere tale configurazione per tutto il periodo considerato. Nella altre quattro run, dove la rete è stata istanziata in modo randomico, in tre casi il nostro modello ha “forzato” tutta la rete ad una configurazione assimilabile ad un paradigma scale free. Imputiamo la non riuscita dell'unico caso (rete randomica, densità 0.1%) alla bassa densità di partenza, che, in un periodo di 1080 iterazioni, non è stato sufficiente a far entrare a pieno regime le meccaniche tipiche di una rete scale-free. C'è ancora margine di miglioramento su questo punto, ma confidiamo nella buona base di partenza offerta dalla nostra soluzione.

Altre considerazioni sono le seguenti:

Omofilia La misura scelta per l'omofilia, ovvero l'*assortatività per attributo*, non sembra aver fornito una risposta soddisfacente sul comportamento omofilo o meno degli agenti. In tutte le run l'indice si mostra parecchio irregolare, e comunque mai completamente positivo. Riteniamo che il problema non sia della metodologia usata per calcolare l'assortatività (piuttosto nota e utilizzata in letteratura), quanto nella rappresentazione dell'attributo di cui questa misura tiene conto: come già spiegato nella sezione 4, abbiamo “condensato” un vettore di interessi in un solo attributo, corrispondente all'interesse con il valore maggiore all'interno del vettore; questo non coglie tutta l'informazione fornita da un vettore di valori, il quale viene usato dagli agenti per stabilire i collegamenti. L'informazione intercettata dall'indice è perciò incompleta. Un possibile miglioramento è senz'altro quello di trovare una misura alternativa, e più esaustiva.

Ipotesi di small-world Anche per l'ipotesi di small-world vale quanto affermato sulla densità reale di un social network come Twitter: è concesso solo supporre il valore reale del cammino minimo medio. In aggiunta, questa è una metrica che risente particolarmente della limitatezza intrinseca della simulazione: con solo 1000 agenti, il cammino minimo medio non può essere paragonato a quello di una rete che dispone di un numero di agenti maggiore di circa cinque ordini di grandezza, tanto più che non sempre in una simulazione si ha a che fare con un grafo连通的, come testimoniato nelle run. Ne concludiamo che l'ipotesi di small-world non possa essere verificata facilmente, se non con un lavoro di analisi su una rete reale.

6.2 Fase 2

In questa fase, proponiamo la simulazione del seguente scenario: un utente “hub” *retwitta* un utente “sconosciuto”. L'obiettivo è studiare la variazione di in-degree dell'utente prescelto. La procedura è molto semplice, e si può riassumere così:

1. Ad una precisa iterazione, viene selezionato l’utente più popolare, ovvero con in-degree massimo, e quello meno popolare, con in-degree minimo.
2. Si crea un tweet “fortunato” e lo si aggiunge alla lista di tweet generale
3. Si crea un retweet da parte dell’utente più popolare contenente il tweet fortunato e lo si aggiunge alla lista generale dei retweet.

Il tweet fortunato è caratterizzato dai seguenti parametri:

- Likability = 1
- Dislikability = 0
- Topic = scelto a caso dalla distribuzioni di topic del nodo “hub”
- Tag = nodo hub

Per le simulazioni abbiamo usato i due modelli migliori in base a quanto visto nella prima fase di esperimenti:

- Modello a invarianza di scala, densità 0.1%
- Modello randomico, densità 1.0%

I parametri della simulazione sono gli stessi della fase 1. L’iterazione prescelta per lo scatto di questo meccanismo è l’iterazione 400; l’abbiamo scelta in quanto lascia al modello 400 iterazioni per “assestarsi”, e 680 iterazioni per studiare il trend dell’in-degree.

L’obiettivo è, come già accennato, modellare il fatto realmente accaduto tra Termite e l’On. Gaspari nel 2012. Come in questo caso, ci aspettiamo che Termite (l’utente sconosciuto) abbia un picco di notorietà negli istanti immediatamente successivi all’evento (tweet fortunato), per poi stabilizzarsi se non addirittura decrescere. Da notare che il picco raggiunto da Termite era di circa 10.000 follower, quindi anche il massimo raggiunto durante le simulazioni dall’utente prescelto dovrebbe essere contenuto rispetto alle dimensioni della rete.

6.2.1 Risultati

Le figure 16, 17 e 18 mostrano i trend per la simulazione della rete scale-free, mentre le figure 19, 20 e 21 mostrano i trend per la rete randomica. La figura 16 fornisce un quadro a grandi linee dell’andamento dell’in-degree della rete a invarianza di scala: durante il periodo di simulazione la maggior parte dei nodi rimane sotto la soglia dei 25 follower, mentre pochi altri spaziano tra questa soglia e i 300 collegamenti in entrata. Ogni nodo sembra manifestare un comportamento unico e non correlato a quello degli altri. In questa situazione abbiamo forzato l’evento di cui sopra. La figura 17 mostra l’in-degree del nodo prescelto, mentre la figura 18 mostra il trend del nodo “hub” che l’ha retwittato.

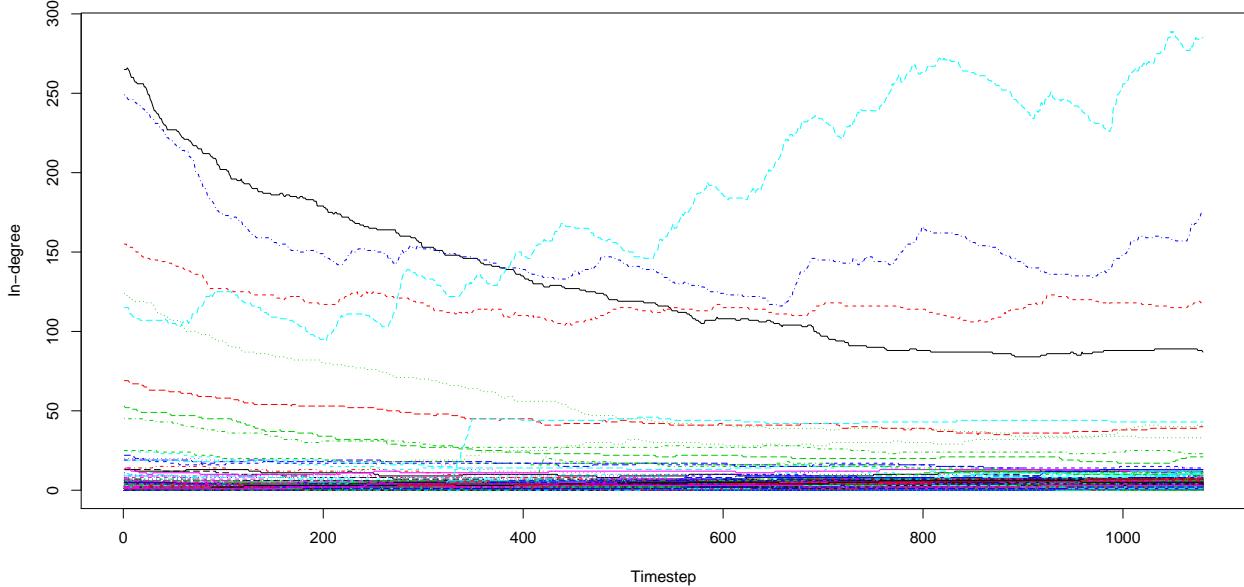


Figura 16: Andamento generale dell'in-degre dei nodi della rete scale-free

Il numero di follower dell’utente prescelto si impenna subito dopo l’iterazione 400, fino a raggiungere il numero di 22 archi entranti. Le iterazioni successive sono caratterizzate da un generico calo dei follower, fino all’assestamento, intorno all’iterazione 950, su circa 15 seguaci. Prevedibilmente, il trend dell’utente popolare non risente del cambiamento, e continua la crescita fino alla terminazione.

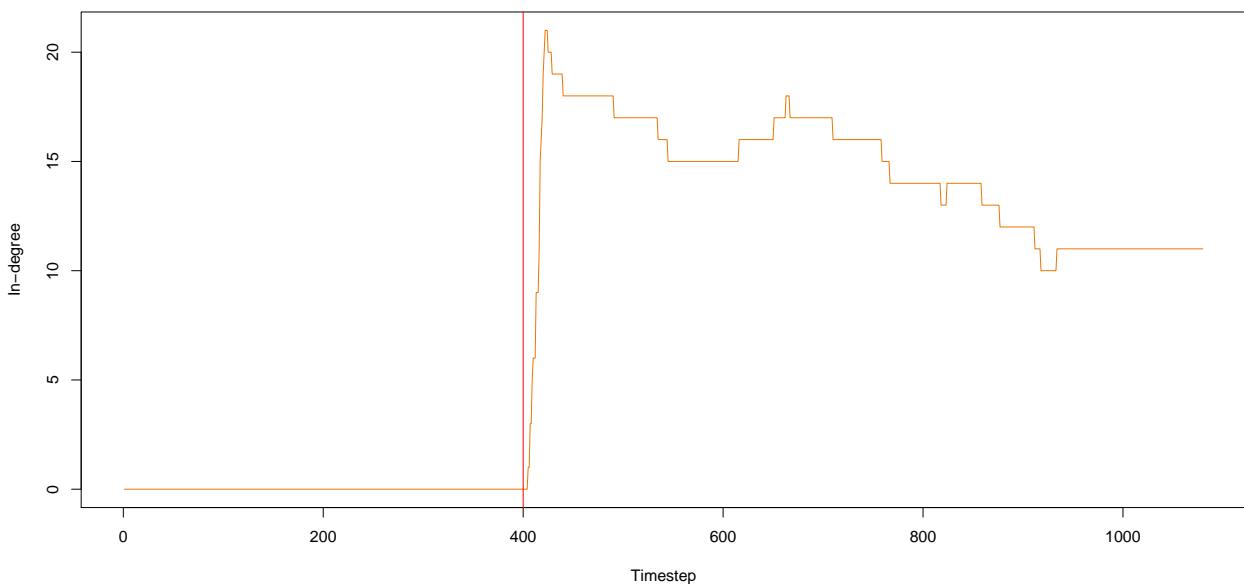


Figura 17: Andamento in-degree del nodo prescelto, nella rete scale-free

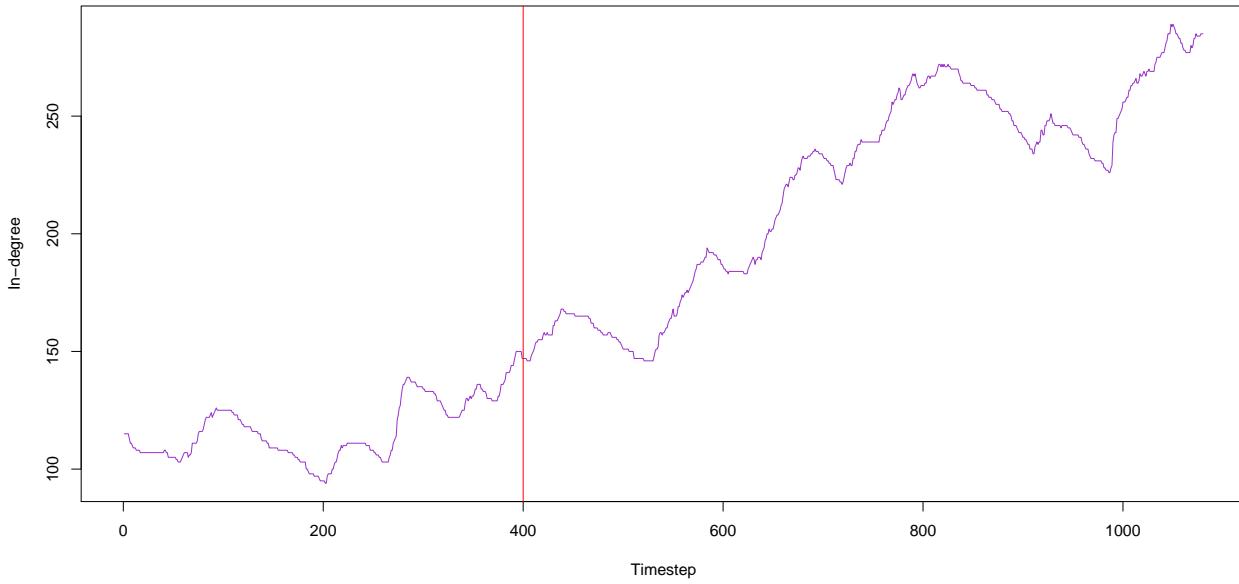


Figura 18: Andamento in-degree del nodo popolare, nella rete scale-free

Una situazione diversa si presenta nella simulazione con rete randomica. La figura 19 mostra un comportamento generale dei nodi piuttosto simile: sebbene la maggior parte rimanga confinata sotto i 100 follower (notare la differenza con la precedente simulazione, dovuta alla diversa densità), gli utenti popolari mostrano quasi tutti un comportamento crescente. Inoltre, questi nodi popolari sono molto più numerosi rispetto alla precedente simulazione. Questo fenomeno dovrebbe essere dovuto al processo descritto nella fase 1, secondo cui la rete randomica si “trasforma” in una rete a invarianza di scala.

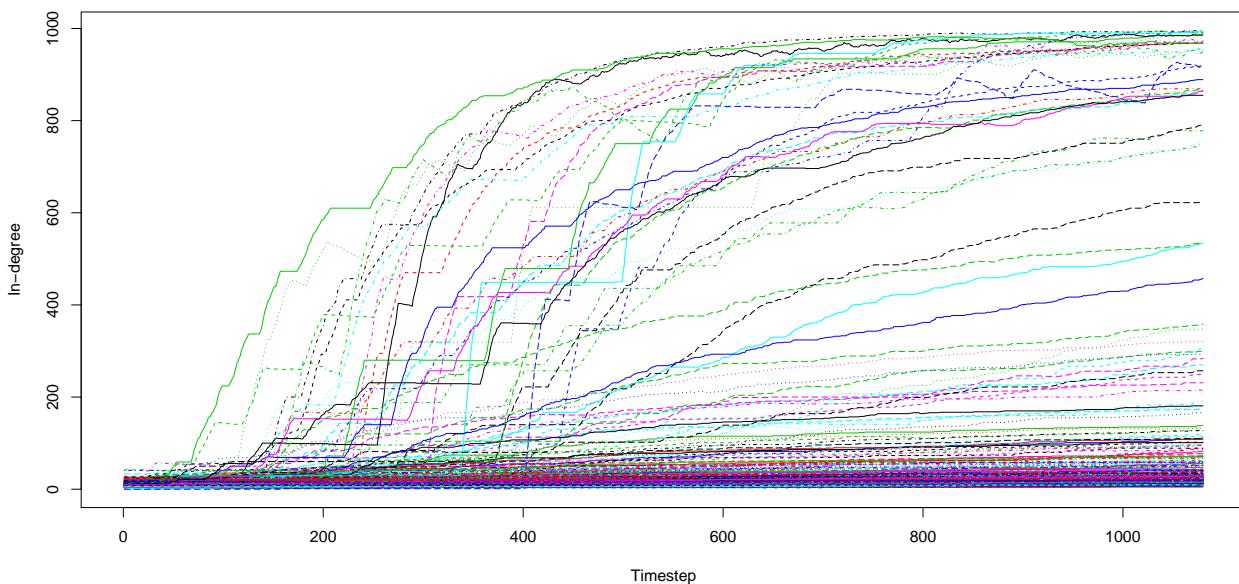


Figura 19: Andamento generale dell'in-degre dei nodi della rete randomica

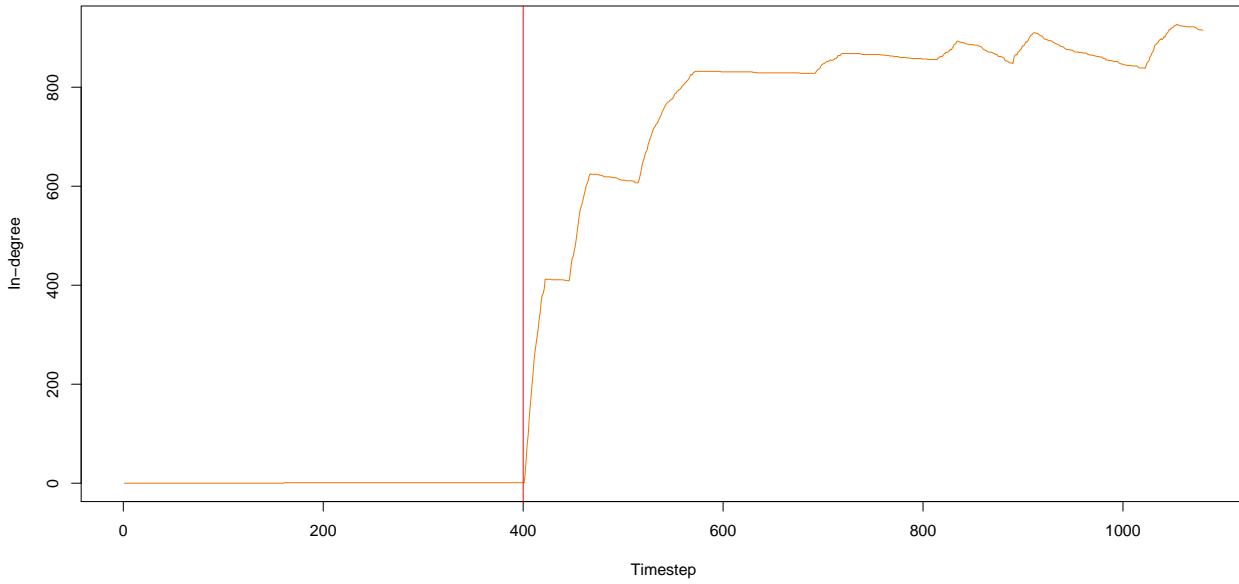


Figura 20: Andamento in-degree del nodo prescelto, nella rete randomica

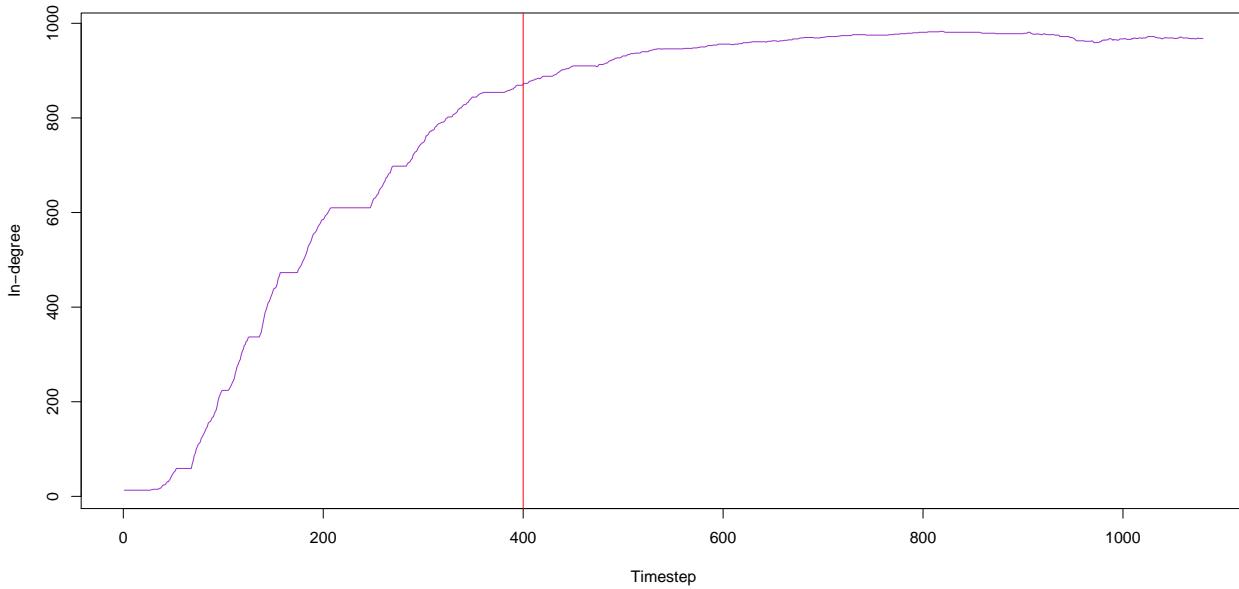


Figura 21: Andamento in-degree del nodo popolare, nella rete randomica

In questo contesto è plausibile un andamento della curva dei follower del nodo prescelto come evidenziato dalla figura 20 l'in-degree, da un valore inferiore a 20, si alza parecchio fino a superare il valore di 800 archi entranti, valore attorno al quale oscilla ma senza mai calare definitivamente. L'utente sconosciuto diventa nell'arco di pochissimo tempo un nodo "hub" a tutti gli effetti. Questo è dovuto sicuramente alla maggiore densità iniziale, che accelera i cambiamenti all'interno della rete, e alla costruzione randomica iniziale, secondo quanto già evidenziato nella fase 1. Da notare anche la differenza di importanza dell'utente famoso tra la

rete randomica e quella scale free: entrambi crescono fino alla fine delle iterazioni, ma mentre la crescita del nodo hub della rete scale-free è molto irregolare e il valore rimane sotto i 300 follower, la crescita della sua controparte in ambiente randomico (figura 21) è molto più regolare, fino al raggiungimento dei 1000 follower, il massimo possibile.

6.2.2 Commento

Dagli esperimenti condotti in questa fase è possibile trarre le seguenti conclusioni:

Rete scale-free Il comportamento generale dei nodi sembra piuttosto “naturale” e veritiero; in aggiunta, dopo il verificarsi dello scenario pianificato, la situazione che si viene a creare è molto simile a quella riscontrata nella realtà: l’utente sconosciuto cresce improvvisamente di popolarità, ma in modo contenuto rispetto alle dimensioni totali della rete. Dopo aver raggiunto il picco, i follower cominciano ad abbandonarlo, fino ad assestarsi su un valore intermedio tra i follower che aveva in precedenza ed il picco raggiunto.

Rete randomica Nonostante il processo di “metamorfosi” da rete randomica a rete scale-free che si verifica durante l’esecuzione del modello, la situazione che si viene a creare sembra non essere adatta alla simulazione di scenari. Il picco di popolarità ottenuto dal nodo prescelto è sproporzionato rispetto a quanto avvenuto nella realtà tra Termite e Gasparri. Probabilmente gli agenti hanno bisogno di più tempo per stabilire trend di popolarità paragonabili a quelli riscontrati nella rete a invarianza di scala. Un fattore importante è sicuramente il limite alla crescita della densità, posto probabilmente troppo in alto (7 – 8%): solo dopo il raggiungimento di questa soglia il comportamento dei nodi può differenziarsi.

7 Conclusione e sviluppi futuri

Abbiamo qui esposto il nostro modello ABSoNeS e gli esperimenti condotti. In linea generale il modello può essere considerato come una buona base di partenza per la simulazione di Twitter, in quanto abbiamo verificato la prevenzione del completamento della rete e la proprietà ad invarianza di scala. In particolare, sia nella fase 1 che nella fase 2 delle prove è emerso come la configurazione iniziale della rete a invarianza di scala sia la migliore per conservarne le proprietà tipiche di una rete sociale e, successivamente, per simulare scenari. Resta il dubbio della densità iniziale, in quanto non è possibile stabilire la densità effettiva di Twitter: la rete a invarianza di scala limita la scelta della densità iniziale allo 0.1%, ma questo può essere aumentato manualmente con archi aggiunti in modo randomico, fino a raggiungere densità di partenza più elevata, seppur inquinando leggermente la power law. In questo modo si ridurrebbe inoltre il numero di nodi con grado 0 presenti all’inizio delle run con bassa densità.

Le reti randomiche hanno il vantaggio principale nella scelta della densità iniziale, ma hanno bisogno di tempo per adattarsi alla soluzione modellistica da noi proposta. Infatti, sono servite a provare la limitatezza della crescita della densità, ma non hanno fornito l’ambiente necessario a simulare uno scenario.

Detto ciò, i possibili miglioramenti per ABSoNeS sono i seguenti:

Densità Scongiurato il problema del completamento della rete, è emerso come il limite posto alla densità sia ancora troppo elevato. Un irrigidimento della probabilità di follow può ovviare a questo problema. In questo modo verrebbe garantita maggiormente la stabilità delle reti e un limite ai “boom” di follower evidenziati soprattutto nelle reti random.

Dimensioni rete Una simulazione di un sistema complesso difficilmente avrà le stesse dimensioni del sistema stesso: per ragioni implementative ne viene preso solo un pezzo o, come nela nostro caso, se ne crea una versione ridotta. È chiaro però che più la simulazione si avvicina come dimensioni al sistema reale, più può essere considerata attendibile; nel caso di social network, il numero di nodi dovrebbe essere quanto più alto possibile. Un possibile sviluppo futuro è dunque quello di aumentare il numero di nodi; un'altra espansione in questi termini è quella di considerare la creazione e cancellazione di nodi. Nella realtà, infatti, nuovi utenti si aggiungono costantemente alla rete, e altri vengono eliminati.

Omofilia Nella fase 1 degli esperimenti abbiamo constatato la limitatezza dell'assortatività per attributo come misura di omofilia. Purtroppo, networkx non ha implementazioni di questa metrica in grado di utilizzare attributi vettoriali. In [21] viene fornita un'espansione della formula di assortatività in grado di misurare l'omofilia tenendo conto di attributi vettoriali: l'implementazione in Python di questa metrica sarebbe un primo passo per il miglioramento dell'analisi di omofilia. Inoltre, il vettore di interessi *PI* può essere sostituito con una rappresentazione più ricca delle caratteristiche e degli interessi di un utente.

Multithreading L'aspetto implementativo sicuramente più interessante è la possibilità di parallelizzare il modello, rendendo la simulazione molto più veloce permettendoci inoltre di aumentare la quantità di nodi ed arco nel grafo iniziale e di permettere un aumento di grandezza dello stesso. Tuttavia non ci è stato possibile seguire questa strada implementativa per via delle restrizioni di Python. Abbiamo implementato il modello *multi-threading* del linguaggio che però risulta essere un modello che non parallelizza veramente il processo in quanto nonostante più di un thread sia attivo sono tutti soggetti al controllo dell'interprete che è single core. La soluzione sarebbe quella di utilizzare il modello *multi-processing* di Python, tuttavia questo richiede la gestione della concorrenza all'accesso e modifica delle strutture dati, e di conseguenza della memoria, relative alla simulazione. Una possibile strada da percorrere potrebbe quella di riscrivere l'intero modello in un linguaggio più a basso livello e pensato per una concorrenza sicura e stabile, come ad esempio **Rust**. Questa transizione porterebbe anche l'invalutabile vantaggio di accellerare notevolmente tutte le operazioni in quanto Rust è pensato per essere una alternativa più moderna e più memory-safe al C++, ed in quanto tale risulta avere velocità paragonabili al C/C++ in quanto è compilato con lo stesso compilatore e utilizza le stesse librerie del C++. Si presuppone un aumento di prestazioni di circa il 130% oltre ad una possibilità di parallelizzare la parte più complessa e costosa in termini di tempo del modello, raggiungendo un aumento di velocità inestimabile. Lo svantaggio più grande sarebbe quello di perdere la possibilità di utilizzare la libreria NetworkX che oltre che essere estremamente ottimizzata contiene diversi parametri essenziali per la valutazione del risultato della simulazione.

8 Disponibilità software

ABSoNeS è disponibile all'indirizzo <https://github.com/Rotuladores/ABSoNeS>. Per il suo funzionamento è richiesto Python 3 e le seguite librerie: NetworkX, NumPy, SciPy e Pandas. Nel repository sono anche presenti le informazioni su come inizializzare ed eseguire la simulazione.

Riferimenti bibliografici

- [1] Welch, Chris (2017). "Facebook crosses 2 billion monthly users". *The Verge*. Vox Media. Retrieved June 27, 2017.
- [2] Bulman, May (2016). "Donald Trump's 'celebrity-style' tweets helped him win US presidential election, says data scientist". *The Independent* Retrieved November 28, 2016.
- [3] Sorkin, Andrew Ross & Peters, Jeremy, W. (2006). "Google to Acquire YouTube for \$1.65 Billion". *The New York Times*. Retrieved October 9, 2006.
- [4] Greene, Jay (2016). "Microsoft to Acquire LinkedIn for \$26.2 Billion". *The Wall Street Journal*. Retrieved June 14, 2016.
- [5] Hamill, L., & Gilbert, N. (2010). Simulating large social networks in agent-based models: A social circle model. *Emergence: Complexity and Organization*, 12(4), 78.
- [6] Barabási, A. L. (2016). Network science. Cambridge university press.
- [7] Barabási, B. A. L., & Bonabeau, E. (2003). Scale-free. *Scientific American*, 288(5), 50-59.
- [8] Ferber, J. (1999). Multi-agent systems: an introduction to distributed artificial intelligence (Vol. 1). Reading: Addison-Wesley. ISO 690
- [9] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, 58(7), 1019-1031.
- [10] Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1), 39-43.
- [11] Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18), 3825-3833.
- [12] McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1), 415-444.
- [13] Currarini, S., Jackson, M. O., & Pin, P. (2009). An economic model of friendship: Homophily, minorities, and segregation. *Econometrica*, 77(4), 1003-1045.
- [14] Zeng, R., Sheng, Q. Z., Yao, L., Xu, T., & Xie, D. (2013, January). A practical simulation method for social networks. In *Proceedings of the First Australasian Web Conference-Volume 144* (pp. 27-34). Australian Computer Society, Inc..
- [15] Singer, H. M., Singer, I., & Herrmann, H. J. (2009). Agent-based model for friendship in social networks. *Physical Review E*, 80(2), 026113.
- [16] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge University Press.
- [17] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). Introduction to algorithms second edition. McGraw-Hill.
- [18] Newman, M. (2010). Networks: an introduction. Oxford university press.
- [19] Hagberg, A., Swart, P., & S Chult, D. (2008). Exploring network structure, dynamics, and function using NetworkX (No. LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Laboratory (LANL). ISO 690

- [20] Foster, J. G., Foster, D. V., Grassberger, P., & Paczuski, M. (2010). Edge direction and the structure of networks. *Proceedings of the National Academy of Sciences*, 107(24), 10815-10820.
- [21] Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, 67(2), 026126.
- [22] Spotti, V (2012). # EPICFAILGasparri, i signori Nessuno e gli eroi del giorno ai tempi di Twitter. www.techeconomy.it
- [23] Bollobás, B., Borgs, C., Chayes, J., & Riordan, O. (2003, January). Directed scale-free graphs. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 132-139). Society for Industrial and Applied Mathematics.
- [24] Russell, S., Norvig, P. (1995). Artificial intelligence. A modern approach. *Artificial Intelligence*. Prentice-Hall, Egnlewood Cliffs, 25, 27. ISO 690