

# Diffusione di una notizia nei Social Network

Gianluca Iselli

Dept. of Engineering and Computer Science  
University of Bologna  
Bologna, Italy  
`gianluca.iselli@studio.unibo.it`

**Abstract.** Questo lavoro è stato svolto come progetto e relazione per l'esame di Sistemi Complessi.

Il lavoro qui presentato è stato ispirato da una notizia della NBC<sup>1</sup> del 30-12-2014. Nell'articolo si racconta di una maestra dell'Oklahoma che ha voluto dimostrare ai suoi studenti, ragazzi di dodicianni, la velocità di diffusione delle fotografie sui social network. La maestra, Melissa Bour, ha condiviso su Facebook un'immagine<sup>2</sup>, che cita "Caro Facebook, [...] i miei alunni di 12 anni pensano "non è nulla di che" pubblicare immagini di loro stessi[...]. Per favore aiutatemi condividendo e commentando con la vostre città di origine. In questa maniera [...] capiranno quanto velocemente le loro immagini possano fare il giro del mondo. [...]".

Successivamente Fox 23 ha ricondiviso, sulla pagina Facebook del giornale, una copia della fotografia della maestra. In un'ora e mezza l'immagine ha raggiunto le 1300 condivisioni ed ottenuto quasi 1000 "mi piace".

Riuscire ad interpretare le proprietà della struttura e degli algoritmi di una rete complessa gioca un ruolo fondamentale in molti campi dei *sistemi complessi*.

Lo studio di queste caratteristiche è spesso richiesto per argomenti importanti come i seguenti:

- Propagazione di una epidemia o pandemie tra la popolazione;
- Diffusione di un virus informatico tra dispositivi mobili e non;
- Divulgazione di una notizia in un piccolo paese, o più in grande tra massmedia e pubblico o, ancora più estesa, in un Social Network

In questo periodo storico i Social Network hanno un successo globale e non è difficile credere che una notizia, su servizi così diffusi, possa fare il giro del mondo.

L'argomento di studio discosta leggermente dalle intenzioni della maestra, infatti, si tratta di un'analisi e di una misurazione di più parametri legati alle reti sociali.

Nell'Introduzione (Capitolo 1) verranno elencati i punti trattati all'interno di questo abstract.

Nel Capitolo "Implementazione" (Capitolo ??) si approfondiranno, poi, le modalità di sviluppo del progetto.

Mentre invece, nel Capitolo "Analisi dei Risultati" (Capitolo 3) sarà presentato quanto elaborato, ovvero il prodotto del lavoro svolto.

---

<sup>1</sup> Notizia sul sito della NBC: <http://www.nbcbayarea.com/news/national-international/Teachers-Social-Media-Lesson-Goes-Viral-on-Facebook-287159381.html>

<sup>2</sup> Immagine condivisa dalla maestra, Melissa Bour: [http://i.dailymail.co.uk/i/pix/2015/01/04/2466352100000578-2895903-image-a-74\\_1420339978179.jpg](http://i.dailymail.co.uk/i/pix/2015/01/04/2466352100000578-2895903-image-a-74_1420339978179.jpg)

## 1 Introduzione

Il capitolo corrente fa da introduzione al lavoro svolto e orienta il lettore ad alcuni aspetti principali che interessano il campo del “Spreading Rumors”.

I primi studi riguardanti la diffusione di una notizia risalgono ai primi anni '60 [2], ma quelli presi in considerazione in questo lavoro sono più attuali e riguardano il campo dei Social Networks.

Lo studio della propagazione di una “voce”, definita più semplicemente notizia o informazione, serve ad analizzare alcuni comportamenti, parametri e modelli di una rete sociale. Le caratteristiche principali sono la topologia della rete e gli utenti che ne fanno parte.

Possiamo quindi iniziare a dire che le simulazioni prevederanno una rete di utenti che “a contatto” con la notizia decideranno a loro volta se ricondividerla o solamente prendere atto della sua esistenza. Al passo 0 della simulazione verrà scelto un utente a cui verrà “insegnata” l’informazione che dovrà essere propagata. Facendo un riferimento al tema della medicina, durante un’indagine epidemiologica, il primo paziente ad aver contratto la malattia viene chiamato paziente zero. Nel nostro caso non si tratta di una malattia ma di una notizia.

Durante le simulazioni verranno inoltre utilizzati alcuni termini che definirò qua di seguito:

- “Ignorants”[1] : al momento della creazione della rete sociale tutti gli utenti vengono definiti in tal modo perchè non sono al corrente della notizia;
- “Spreaders”[1] : tutti gli utenti che decidono di condividere la notizia. Il “paziente zero” fa parte di questo gruppo;
- “Uninterested” : tutti quegli utenti che dopo essere diventati consapevoli dell’esistenza della notizia decidono comunque di non condividerla;
- “Viewers” o visualizzatori : tutti coloro facenti parte del gruppo degli Spreaders e degli Uninterested.

Facendo una veloce panoramica, l’argomento di questo progetto è certamente un ottimo ambiente di studio ed il numero esorbitante di utilizzatori di Social Networks in circolazione crea sicuramente un terreno fertile per la diffusione di notizie ed informazione.

### 1.1 Obiettivi

Per questo lavoro possiamo definire tre obiettivi principali.

Il primo test permetterà di decidere il modello topologico più consono, tra quelli presi in considerazione, per i successivi test.

Il secondo servirà per mettere in luce come una notizia, con un topic adatto maggiormente ad una certa fascia d’età, venga diffusa nei differenti Social Networks. E’ presente una statistica online piuttosto recente (3°trimestre 2014) che mostra la distribuzione per età di come è divisa l’utenza nei social networks più popolari. Il grafico in figura 1 ne mostra la distribuzione.

L’ultimo obiettivo, invece, servirà ad analizzare un’interazione tra 2 diversi gruppi di utenti, che permetterà di studiare il numero di visualizzazioni della

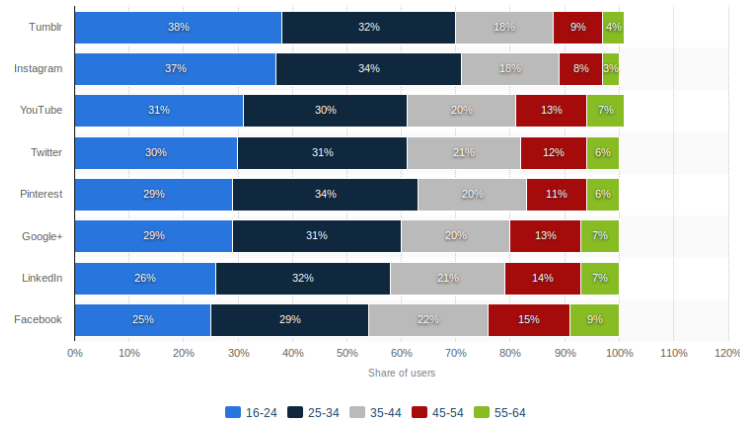


Fig. 1: Distribuzione delle età divisa per Social Network [5]

notizia in casi più complessi. Un esempio potrebbe essere quello di voler dividere il numero totale delle persone in due sottoinsiemi così formati:

- Il primo gruppo è formato da pochi utenti, tipo il 20% del totale, ma ogni componente ha un'ottima probabilità di condivisione.
- Il secondo gruppo, viceversa, è formato dall'80% degli utenti, ma ogni componente ha una possibilità minore di condivisione.

## 2 Progettazione e Implementazione

Per favorire un modello da studiare che permetta di risolvere gli obiettivi descritti pocanzi sono state prese le decisioni qui di seguito elencate.

Iniziamo con il dire che il social network verrà astratto ad un grafo scale-free dove ogni nodo è un utente che possiede alcune caratteristiche.

Per il primo obiettivo verrà posta l'attenzione su due algoritmi, impiegati per la creazione di grafi, che formano modelli di rete differenti.

Inizialmente verrà fatto vedere come una notizia viene propagata in un grafo di tipo "Preferential Attachment" suggerito da Barabási e Albert [3].

Le simulazioni poi proseguiranno con un'altra topologia di grafo, sempre scale-free con Power Law Degree, descritta però dall'algoritmo di Dorogovtsev e Mendes [4].

Il lavoro non parte da dati reali e la scelta di queste topologie di grafi è data dalla peculiarità di alcune loro caratteristiche che elencherò qui di seguito:

- La topologia di grafo Preferential Attachment:
  - è un modello ampiamente utilizzato per la sua semplicità;
  - è impiegato da buona parte di studi che trattano argomenti somiglianti lo "Spreading Rumors";

- non è un grafo di tipo frattale [6] ma ha una caratteristica simile. Non possedendo cricche di almeno 3 nodi, se un nodo non condivide la notizia, tutti i suoi nodi “figli” non riceveranno mai l’informazione.
- La topologia di grafo definita da Dorogovtsev e Mendes invece:
  - ha un modello decisamente più complesso;
  - può avere all’interno del grafo cricche da 3 o più nodi e questo permette una probabilità maggiore di condivisione della notizia;
  - più somigliante<sup>3</sup> ad una struttura reale di rete sociale.

Cercando di migliorare il modello totale della simulazione è stato deciso di fornire alcune proprietà alla notizia e agli utenti(Nodi) che compaiono nella rete sociale.

Osservando il secondo obiettivo dobbiamo fornire alla notizia un “argomento” e per fare ciò si è optato per l’inserimento di  $N$  valori che definiranno quanto è adatta la notizia per ogni fascia d’età.

Avendo a disposizione la distribuzione delle età, precedentemente mostrata nel grafico di figura 1, possiamo notare 5 intervalli di anni e quindi stabilire il valore di  $N = 5$ .

Dobbiamo perciò anche definire 5 gruppi di utenti per distinguere le differenti età. Il numero di persone appartenenti ad ogni gruppo verrà semplicemente definito dalla semplice proporzione:

$$\frac{N\_NODI\_TOTALI}{100} \cdot \%\_UTENTI\_GRUPPO$$

dove  $\%\_UTENTI\_GRUPPO$  è la percentuale presente nel grafico di figura 1.

A questo punto per la condivisione dell’informazione tra nodo e nodo manca solamente la formula per definire la probabilità della propagazione. L’idea è quella di dare la “possibilità” all’utente, come nella realtà, di decidere se la notizia gli interessa oppure no. Questo ci obbliga ad aggiungere un nuovo parametro all’utente, l’astensione alla notizia.

Così facendo la notizia verrà condivisa dall’utente ‘*i-esimo*’ con una certa età ‘*e*’ se:

$$NOTIZIA_e > UTENTE_i.astensione$$

dove  $NOTIZIA_e$  è la forza di condivisione della notizia su una data fascia d’età ‘*e*’, mentre  $UTENTE_i.astensione$  è l’astensione alla notizia dell’utente *i*-esimo.

L’ultimo obiettivo agisce proprio su quest’ultima parte appena definita. Quando si parla di “forza della notizia” e di “forza di astensione” vogliamo considerare un numero tra 0 e 1. Per avere una probabilità di condivisione più o meno alta andremo ad agire solo sulla “forza di astensione”. Per come abbiamo definito la formula della condivisione basterà che la distribuzione dell’astensione sia più tendente a 0 per avere una miglior condivisione, o viceversa, più tendente a 1 per avere una peggior condivisione.

<sup>3</sup> Non esiste un modello virtuale di una rete sociale reale. Tutte le ottimizzazioni che vengono apportate sono per rendere i risultati di queste simulazioni più attinenti alla realtà.

## 2.1 Strumenti

Dopo aver discusso di tutte le fasi della progettazione possiamo passare all'implementazione.

Le simulazioni previste saranno sviluppate in NetLogo che fornisce un ambiente semplice e piuttosto personalizzabile dove si possono condurre studi in parecchi campi. NetLogo è uno strumento di sviluppo per una programmazione basata ad agenti. Nel lavoro qui presentato gli agenti in questione saranno gli utenti della rete sociale. Esistono molti altri simulatori di reti che prevedono una programmazione ad agenti, ma NetLogo ha risposto brillantemente alle esigenze implementative avendo una sintassi semplice e una moltitudine di API<sup>4</sup> [7] fornite dal linguaggio stesso. E' stato anche scelto per la facilità con cui si riesce a modellare una interfaccia grafica<sup>5</sup>.

## 2.2 Topologie dei grafi

Come già annunciato all'inizio di questo capitolo i modelli di grafo che verranno studiati in questo lavoro sono due.

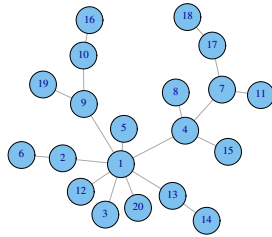
**Preferential Attachment** Il modello Preferential Attachment, come già comunicato a inizio capitolo, è altamente diffuso e conosciuto. Tanto da avere una propria API per la creazione nel dizionario di NetLogo.

Invocando la funzione:

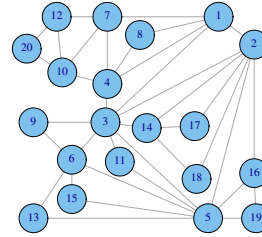
```
| generate-preferential-attachment turtles links num-nodes
```

che risiede all'interno dell'estensione "network" verrà creato un grafo di *num-nodes* nodi con le proprietà del grafo Preferential Attachment. L'algoritmo di PA implementato in NetLogo crea per ogni nuovo nodo un solo collegamento.

In figura 2a è presente una rappresentazione grafica di un grafo con 20 nodi.



(a) Preferential Attachment



(b) Dorogovtsev e Mendes

Fig. 2: Rappresentazione grafica delle due topologie di grafo studiate.

<sup>4</sup> Application Programming Interface: un insieme di funzionalità utilizzabili dal programmatore

<sup>5</sup> Dall'inglese GUI ovvero Graphical User Interface

**Rete di Dorogovtsev e Mendes** Il modello definito da Dorogovtsev e Mendes viene implementato dalla libreria Java chiamata GraphStream [8]. La libreria è formata essenzialmente da tre parti:

- core: Il package principale di GraphStream;
- algo: Il package dove sono implementati tutti gli algoritmi e i generatori della libreria;
- ui: Il package che permette di visualizzare e dare un layout al grafo.

I grafi che verranno utilizzati nelle simulazioni sono stati creati da un'applicazione implementata appositamente. Una volta creati, questi grafi vengono salvati su file in formato GraphML [9] per dar modo di essere caricati dalla simulazione di NetLogo.

In figura 2b è presente una rappresentazione grafica di un grafo con 20 nodi.

### 2.3 Inizializzazione del Modello

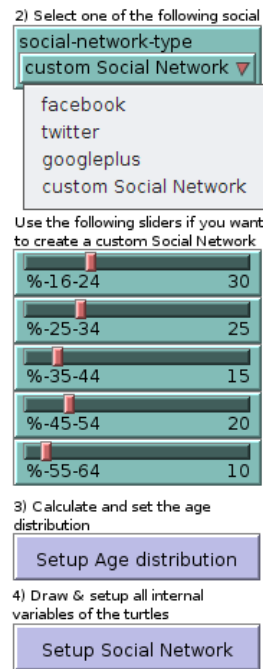


Fig. 4: GUI: scelta del Social Networks; pulsante per il calcolo delle età; pulsante per l'inizializzazione dei nodi.

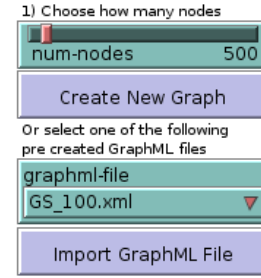


Fig. 3: GUI: Creazione o importazione del grafo

Al fine di portare a termine gli obiettivi in precedenza descritti è stata creata una interfaccia grafica per permettere all'utilizzatore di interagire con il progetto. Nella figura 3 e nella figura 4 vengono mostrate due porzioni di GUI che servono all'inizializzazione delle simulazioni. Nella maggior parte delle dimostrazioni di NetLogo è presente solo un pulsante "Setup" e un pulsante "go". Data la particolarità del progetto non è stato possibile semplificare a questo livello l'interfaccia grafica perciò, qui di seguito, verranno illustrate le caratteristiche principali.

In figura 3 si possono notare due pulsanti:

- il primo serve per creare un grafo di tipo Preferential Attachment (Capitolo 2.2) con un numero *num-nodes* di nodi.
- il secondo invece serve per importare un file GraphML (Capitolo 2.2) tra quelli già creati in precedenza.

Dopo aver creato o importato il grafo, viene cercato l'hub della rete, ovvero il nodo con il *degree* più alto.

Un'altra fase necessaria è stata quella di importare nel progetto i dati di alcuni Social Networks presenti nel grafico di figura 1.

Sono stati inclusi solo i tre che vengono mostrati in figura 4 per via della loro popolarità. In alcune simulazioni esiste la necessità di creare condizioni particolari che non possono essere risolte con i dati di questi Social Networks. Per rendere l'applicazione più generale possibile sono stati implementati anche 5 cursori che permettono di far decidere all'utilizzatore dei parametri personalizzati. Ognuno di questi cursori serve per impostare la percentuale di utenti facente parte della suddetta fascia d'età. Per semplicità sono stati chiamati:

“%-<fascia età inizio>-<fascia età fine>”.

Si può vedere la loro implementazione nell'esempio in figura 4 dove si nota che:

- il gruppo formato dalle persone che hanno tra i 16 e i 24 anni sono il 30%;
- quelle che hanno tra i 25 e i 34 anni sono il 25%;
- quelle che hanno tra i 35 e i 44 anni sono il 15%;
- quelle che hanno tra i 45 e i 54 anni sono il 20% gruppo;
- quelle che hanno tra i 55 e i 64 anni sono il 10%;

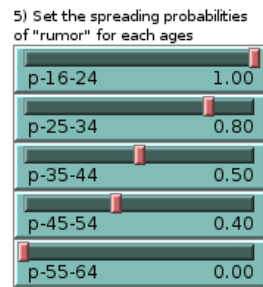


Fig. 5: GUI: cursori per la definizione di una notizia.

Per calcolare quanti utenti devono essere presenti per ogni intervallo di età bisogna premere il pulsante “Setup Age Distribution”. Verrà creato un vettore grande come il numero dei nodi del grafo e al suo interno saranno inseriti gli indici dei 5 gruppi nella quantità prevista dalle percentuali. Questo farà sì che ogni nodo del grafo avrà un suo preciso valore nell'array che identificherà il gruppo a cui lui appartiene. Il pulsante “Setup Social Networks”, invece, serve a resettare i dati base degli utenti, riportando la simulazione allo stato iniziale, senza eliminare né il grafo né il vettore delle “età”. Per stato iniziale della simulazione si intende impostare nell'hub la notizia, facendolo diventare uno

“Spreader” mentre tutti gli altri nodi del grafo verranno fatti diventare “Ignorant” e per ciascuno sarà calcolato un nuovo valore di astensione alla notizia.

Sono presenti altri 5 cursori, mostrati in figura 5, che identificano l'intensità con cui la notizia è forte verso un certo gruppo di utenti. Viene definita come

“p-<fascia età inizio>-<fascia età fine>” per indicare la probabilità di condivisione che la notizia ha sulla particolare fascia d'età. Nell'esempio in figura 5 la probabilità nel primo gruppo vale 1.0, ovvero la condivisione avrà *sempre* esito positivo. Con il quinto

gruppo avremo un caso contrario dove la probabilità è 0 e la condivisione non andrà mai a buon fine. Nei casi intermedi viene confrontato il valore di astensione dell'utente al valore del cursore dell'età dell'utente come è stato definito in fase di progettazione (Capitolo 2).

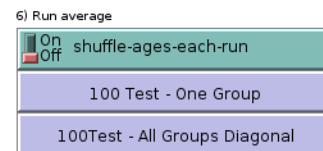


Fig. 6: GUI: pulsanti per dare il via ai primi due test

I pulsanti presenti in figura 6 servono a dare il via al test del primo e secondo obiettivo.

## 2.4 Sviluppo delle Simulazioni

Oltre all’inizializzazione della rete e alla configurazione dei suoi agenti, è stata progettata un’architettura in grado di soddisfare gli obiettivi prefissati.

**Propagazione della Notizia** L’algoritmo 1, data la sua importanza, è stato definito il nucleo delle simulazioni ed è colui che calcola, ad ogni passo, come la notizia viene propagata.

---

### Algorithm 1 Nucleo della propagazione della notizia

---

```

1: procedure CORE(nodes, news)
2:   if (all nodes have seen the news) then
3:     return END_OK
4:   else
5:     for all (node in nodes with [seen_news = false]) do
6:       if (any nodeneighbors with [shared_news = true]) then
7:         ▷ Leggo il valore di probabilità della notizia per l’età del utente corrente
8:         p_news  $\leftarrow$  newsprob[nodeage]
9:         if p_news  $\geq$  node.chance_of_abstention then
10:          nodetmp_shared_news  $\leftarrow$  true
11:        end if
12:        nodetmp_seen_news  $\leftarrow$  true
13:      end if
14:    end for
15:    if (no nodes with [tmp_seen_news = True]) then
16:      return END_KO
17:    end if
18:    for all (node in nodes with [tmp_seen_news = true]) do
19:      nodeseen_news  $\leftarrow$  true
20:      nodetmp_seen_news  $\leftarrow$  false
21:    end for
22:    for all (node in nodes with [tmp_shared_news = true]) do
23:      nodeshared_news  $\leftarrow$  true
24:      nodetmp_shared_news  $\leftarrow$  false
25:    end for
26:    return SIMULATION_NOT_FINISHED_YET
27:  end if
28: end procedure

```

---

Prima verifica se ci sia qualche nodo che non ha visualizzato la notizia (Riga 2), poi richiede a tutti i nodi “Ignorants” se hanno un vicino “Spreader” (Riga 5 e 6). In tal caso viene letta la probabilità della notizia rispetto all’età del



nodo “Ignorant” corrente (Riga 8) e viene eseguito il confronto con il valore di astensione alla notizia (Riga 9), come stabilito in fase di progettazione descritta nel capitolo 2. Esiste un'altra funzione altrettanto importante e si occupa di fare la media di  $N$  test del numero di “Viewers”. Ad ogni esecuzione però si preoccupa anche di effettuare il reset di tutti gli agenti che prendono parte alla simulazione calcolando, nuovamente, un valore di astensione per ogni nodo e, se richiesto, viene anche rimescolato il vettore che mantiene le età per avere più casualità nei test effettuati.

**“100 Test - One Group”** Questo pulsante viene premuto per eseguire la prima simulazione. Come prima azione verrà impostato il Social Network di tipo “custom”, che permetterà di poter personalizzare la percentuale di utenti nei 5 gruppi, impostando il 100% nel primo gruppo e 0% negli altri gruppi.

Successivamente il test eseguirà la procedura di propagazione della notizia facendo variare la probabilità della notizia per il primo gruppo, quindi il cursore “p-16-24”, da 0 a 1 con passi di 0.05. Si arriverà ad avere un grafico che mostrerà sull'asse X i ticks, ovvero l'avanzamento della simulazione, quindi la forza della notizia. Mentre sull'asse Y ci sarà la media di 100 test del numero di “Viewers”.

**“100 Test - All Groups Diagonal”** La seconda simulazione utilizza invece le percentuali di utenti definite nel grafico della statistica di figura 1 facendo variare una delle probabilità della notizia da 0 a 1 con step di 0.05. Inizialmente verrà scelto di studiare cosa capita se si fa variare solo la forza della notizia del gruppo più numeroso. Successivamente verrà invece analizzato come si comporta la rete quando si fa variare la forza della notizia di tutti i gruppi. Così facendo si arriverà ad avere un grafico dove sull'asse X ci saranno nuovamente i ticks, ovvero l'avanzamento della simulazione, quindi la forza della notizia. Mentre sull'asse Y ci saranno tutte le medie di 500 test del numero di “Viewers” di ogni gruppo.

**“Great Sharers vs. Little Sharers”** L'ultima simulazione è quella che si distingue di più dalle altre dato il concetto che ci sta dietro. A differenza dei precedenti test, si vuole studiare il confronto tra due gruppi che si differenziano per la loro probabilità di astensione alla notizia. Nelle prime due simulazioni questo dato veniva deciso da una funzione random che possiede una distribuzione di densità lineare. In questo test, invece, l'astensione viene calcolata tramite una funzione di distribuzione di probabilità non lineare. Quella scelta è la funzione Weibull[10]. La funzione “random-weibull” fa variare la densità della distribuzione per mezzo di due parametri  $\alpha$  e  $\beta$ . Il valore restituito dalla funzione è sempre un numero tra 0 ed 1 perciò, per il calcolo

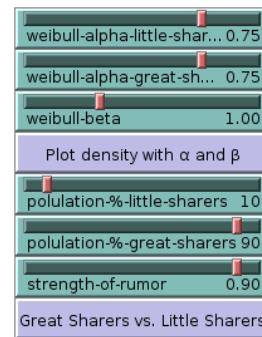


Fig. 7: GUI: terzo test

1 perciò, per il calcolo

della propagazione della notizia, può essere riutilizzato l'algoritmo 1 mostrato in precedenza. In figura 7 sono presenti 3 slider per la manipolazione degli  $\alpha$  e di  $\beta$ . Il parametro  $\beta$  fa modificare soprattutto la forma della curva, mentre  $\alpha$  la pendenza. Quello che ci serve è una pendenza differente perciò sono stati aggiunti due cursori per i parametri  $\alpha$  dei due gruppi. In figura 8 viene mostrato il dislivello di cui si sta parlando. Nel primo istogramma, con  $X$  più densa vicino a 0, viene permessa una miglior condivisione mentre nel secondo grafico se ne ha una peggiore.

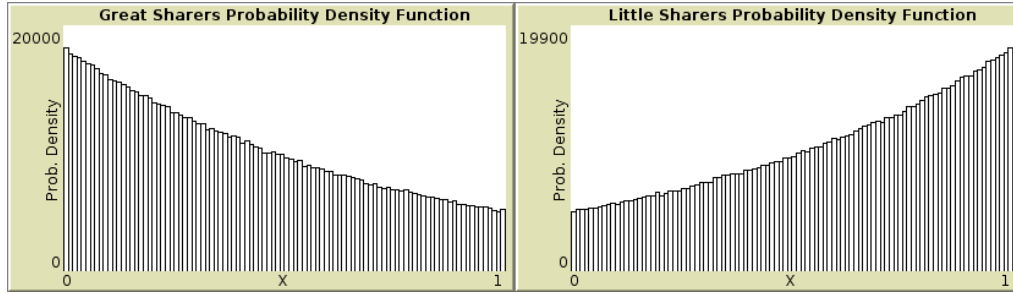


Fig. 8: I due istogrammi mostrano la rappresentazione di un vettore  $X$  di 1 milione di cifre calcolato mediante la funzione di densità di probabilità Weibull, con  $\alpha = 0.75$ . Il grafico di sinistra rappresenta la distribuzione originale, mentre quello di destra la sua *quasi*<sup>6</sup> speculare calcolata grazie:  $\forall x \in X, \quad 1 - x$ .

In figura 7 sono presenti altri 2 cursori per definire la percentuale degli utenti appartenenti ad ogni gruppo e il cursore per impostare la “forza” della notizia. Alla pressione del tasto “Great Sharers vs. Little Sharers” viene lanciata una simulazione che riguarda i parametri selezionati. Vista la quantità di test da fare con tutti le combinazioni di parametri possibili è stata scritta una procedura<sup>7</sup> che, grazie ai suoi tre cicli annidati, riesce a considerare tutte le possibilità richieste dal terzo obiettivo. L'algoritmo 2 mostra su quali parametri agiscono i tre cicli.

Il primo ciclo, più esterno, itera su alcuni valori di forza della notizia, ovvero i seguenti: 0.50 0.60 0.70 0.75 0.80 0.90.

Il secondo ciclo, quello centrale, modifica il numero di individui per ogni gruppo. Ad ogni passo viene incrementato il numero di persone con un'alta possibilità di condivisione passando dal 10% al 90%. Operazione inversa viene eseguita per le persone con bassa possibilità di condivisione passando dal 90% al 10% della popolazione totale. Il terzo ciclo, più interno, invece itera sul parametro  $\alpha$  modificando quindi la pendenza della curva per gli individui con alta probabilità

<sup>6</sup> Quando si parla di numeri random non possiamo pretendere di avere lo stesso identico comportamento.

<sup>7</sup> La procedura che permette di fare tutti i test impiega un tempo prolungato per la sua esecuzione. Per invocarla basta inserire il comando “great-sharers-in-comparison-to-little-sharers-all-tests” in modalità *Observer* nel *Command Center* di NetLogo.

di condivisione. Come si può notare il parametro  $\alpha$  del gruppo che condivide meno viene preso in ingresso dalla funzione e mai manipolato.

---

**Algorithm 2** Tre cicli annidati per considerare tutte le possibilità interessanti del terzo obiettivo

---

```

1: procedure THIRD-TEST-ALL(alpha_little_sharers)
2:   weibull_alpha_little_sharers  $\leftarrow$  alpha_little_sharers
3:   for all (news_strength in [0.50 0.60 0.70 0.75 0.80 0.90]) do
4:     polulation_%_step  $\leftarrow$  10
5:     while (polulation_%_step  $\leq$  90) do
6:       %_little_sharers  $\leftarrow$  (100 - polulation_%_step)
7:       %_great_sharers  $\leftarrow$  polulation_%_step
8:       for all (weibull_alpha_great_sharers in [0.25 0.50 0.75 1.0]) do
           $\triangleright$  Eseguo la simulazione 500 volte con i parametri correnti per avere un
          risultato mediato più valido possibile
9:       end for
10:      polulation_%_step  $\leftarrow$  polulation_%_step + 5
11:    end while
12:  end for
13: end procedure

```

---

### 3 Analisi dei Risultati

Prima di iniziare ad elencare i test effettuati e mostrare i risultati ottenuti, è necessario sottolineare una questione relativamente importante. Dopo aver visto come si comporta l'applicativo con grafi molto grandi ( $> 500$  nodi) è stato ottenuto opportuno abbassare il numero dei nodi affinché l'esecuzione non portasse via troppo tempo e risorse. Tutte le analisi sono state effettuate con reti composte da 500 nodi. Un secondo punto da chiarire è che avendo due applicativi, il primo in Java per creare grafi (Capitolo 2.2) e il secondo in NetLogo per utilizzarli nelle simulazioni, è stato ritenuto opportuno non cambiare mai il grafo durante i test.

#### 3.1 Primo test

In questa prima analisi verranno confrontati i due tipi di grafo descritti nel capitolo 2.2.

Per questo test abbiamo bisogno di:

- un gruppo di utenti che forma il 100% dei nodi;
- un grafo di 500 nodi creato grazie all'algoritmo Preferential Attachment (Capitolo 2.2);
- un grafo di 500 nodi creato grazie all'algoritmo di Dorogovtsev e Mendes (Capitolo 2.2);
- un ciclo che vari la “forza” della notizia portandola da 0 a 1 con passi di 0.05.

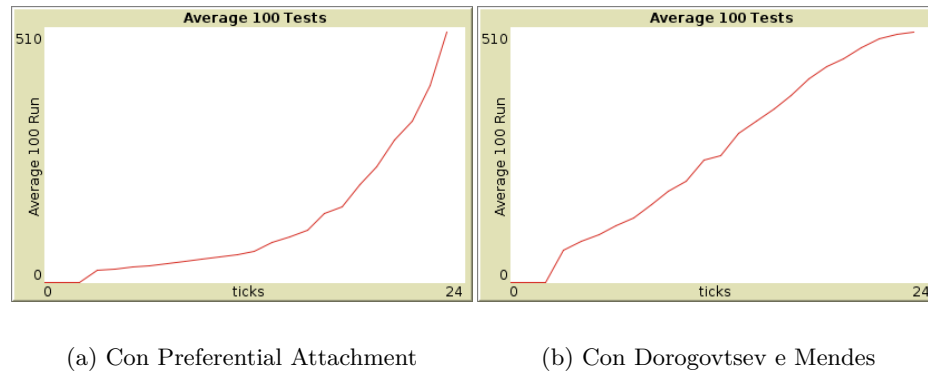


Fig. 9: Risultati del primo test

Con il grafo descritto da Barabási e Albert si ottiene il risultato che viene mostrato nel grafico di figura 9a. Come si nota ha una crescita molto lenta e solo dopo che la probabilità di condivisione ha superato il valore di 0.5 si ha

una salita tangibile. La topologia descritta da Dorogovtsev e Mendes invece si comporta molto meglio e viene mostrata nel grafico in figura 9b. Ha una crescita quasi lineare e alla fine, quando la forza è attorno l'85% crea una “pancia” verso l'alto che sta a dimostrare l'altissima condivisione.

Il primo grafo è molto svantaggiato per via della sua struttura. Come abbiamo detto in fase di progettazione (Capitolo 2) questo grafo non presenta cricche al suo interno. Questa proprietà lo penalizza nelle simulazioni che sono basate sul numero di visualizzazioni e perciò sulla probabilità di condivisione. Nel modello descritto da Dorogovtsev e Mendes invece sono presenti cricche al suo interno ed il grafo di questo esempio ne è la prova avendo quasi il doppio dei link del Preferential Attachment.

Com'è stato appena esposto, il confronto tra i due modelli di grafo ci ha permesso di assumere che la seconda topologia affronti meglio questo studio. Si è perciò optato di sostenere i prossimi test con la seconda topologia realizzata da Dorogovtsev e Mendes.

Inoltre, anche se è solo un punto di vista dell'autore, si può notare in questo modello di grafo una maggior somiglianza con la topologia più comune di un Social Network.

### 3.2 Secondo test

In seguito al confronto studiato nel primo test, ed aver deciso quale fosse il modello di grafo migliore per le analisi, possiamo concentrarci sulla seconda fase definita negli obiettivi.

Questo secondo studio ci permetterà di analizzare come viene condivisa una notizia, con un argomento, viene propagata nei differenti social.

vuole mettere in luce come una notizia, con un topic adatto maggiormente ad una > > certa fascia d'età, viene propagata nei differenti social.

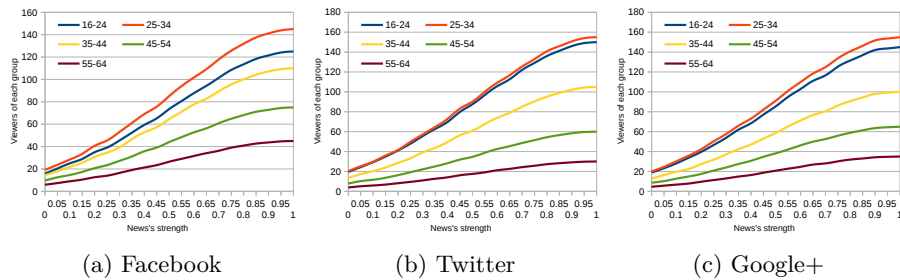


Fig. 10: Risultati del secondo test

### 3.3 Terzo test

In quest'ultima analisi, invece, verrà mostrata un'interazione tra 2 diversi gruppi di utenti (Nodi). Il primo gruppo è formato da persone con una maggior probabilità di condividere la notizia mentre nel secondo, al contrario, da persone con una minor probabilità. Questo studio punta ad analizzare quante visualizzazioni vengono fatte per una singola informazione condivisa. Verrà anche condotto uno studio in cui il totale degli utenti si divide in due gruppi costituiti da, ad esempio, pochi utenti con alte probabilità di condividere l'informazione e, viceversa, molti utenti con basse probabilità di condivisione.

La possibilità di condivisione è data nuovamente dal confronto tra la “forza della notizia” e la “forza di astensione”. In questo caso, però, l'astensione viene calcolata tramite una funzione di distribuzione di probabilità non lineare, quella scelta è la funzione Weibull già descritta al capitolo 2.4.

Per ottenere una curva di distribuzione più o meno inclinata, in seguito ai tentativi attuati, è stato ritenuto opportuno mantenere il valore di  $\beta$  costante a 1.0 e variare, invece, quello di  $\alpha$ .

Il valore di  $\alpha$  per il secondo gruppo invece è stato fissato a 0.75 perchè la distribuzione di figura 8 mostra un dislivello non troppo alto che consente, comunque, un buon grado di condivisione della notizia.

Riassumendo il test viene composto da i parametri dinamici citati nel capitolo 2.4 e dai seguenti parametri statici:

- La dimensione della popolazione non cambia mai e resta sempre di 500 Nodi;
- Il parametro  $\alpha$  del gruppo con bassa probabilità di condivisione rimane 0.75, ottenendo una densità di probabilità come in figura 8;
- La topologia del grafo è la stessa per ogni esecuzione del test;

La figura 11 mostra il risultato atteso, ovvero la crescita del numero di visualizzazioni sia al variare di  $\alpha$  che all'aumentare dei nodi con alta probabilità di condivisione.

Si vuole inoltre porre l'attenzione su come con un valore di  $\alpha$  molto basso (0.2) ed un numero di nodi con alta probabilità di condivisione pari a 150 corrisponda a 300 visualizzazioni totali, quasi le stesse ottenute da un valore di  $\alpha = 1.0$  e 325 nodi con alta probabilità di condivisione.

## 4 Conclusioni

In conclusione i risultati posti come obiettivo sono stati raggiunti.

è stato fatto vedere questo dell'obiettivo e ha avuto questa risoluzione, poi questo con questo e bla bla bla

Come abbiamo potuto osservare sono pienamente in linea con

### 4.1 Sviluppi Futuri

Prendere i risultati del 3° test e confrontarli con quelli dei test base e vedere come si comportano, minore? maggiore?

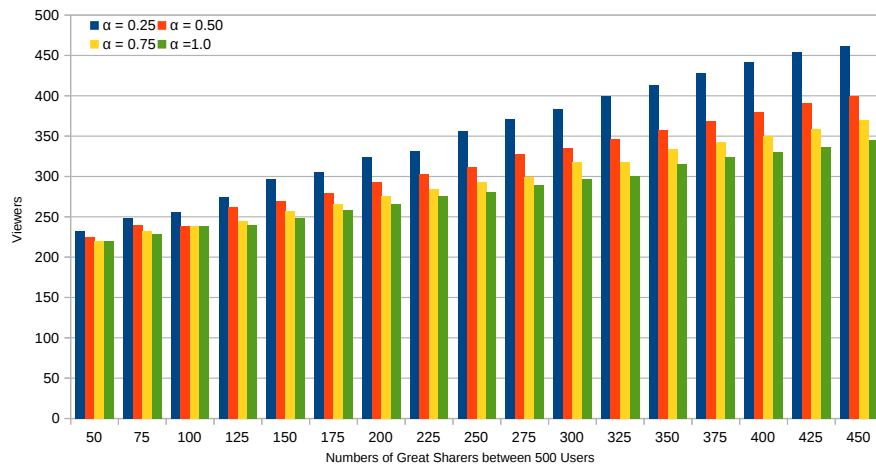


Fig. 11: Grafico del risultato dell'ultimo test con forza della notizia pari a 0.50, 500 nodi totali e  $\alpha$  del gruppo con bassa probabilità di condivisione = 0.75

## References

1. Nick Fedewa, Emily Krause, Alexandra Sisson and Advisor: James Angelos.  
*Spread of A Rumor.*  
<http://www.siam.org/students/siuro/vol6/S01182.pdf>
2. D. J. Daley, D. G. Kendall.  
*Stochastic Rumours.*  
<http://imamat.oxfordjournals.org/content/1/1/42.abstract>
3. A.-L. Barabási, R. Albert.  
*Emergence of scaling in random networks*  
[http://www.barabasilab.com/pubs/CCNR-ALB\\_Publications/199910-15\\_Science-Emergence/199910-15\\_Science-Emergence.pdf](http://www.barabasilab.com/pubs/CCNR-ALB_Publications/199910-15_Science-Emergence/199910-15_Science-Emergence.pdf)
4. S. N. Dorogovtsev, J. F. F. Mendes.  
*Evolution of networks*  
<http://arxiv.org/pdf/cond-mat/0106144v2.pdf>
5. Statista 2015.  
*Age distribution of active social media users worldwide as of 3rd quarter 2014, by platform .*  
<http://www.statista.com/statistics/274829/age-distribution-of-active-social-media-users-worldwide-by-platform/>
6. Zhongzhi Zhang, Shuigeng Zhou, Tao Zou, Guisheng Chen.  
*Fractal scale-free networks resistant to disease spread.*  
<http://arxiv.org/pdf/0804.3186.pdf>
7. <http://ccl.northwestern.edu>.  
*NetLogo Dictionary.*  
<http://ccl.northwestern.edu/netlogo/docs/dictionary.html>
8. GraphStream Team  
*GraphStream Project.*  
<http://graphstream-project.org/>

9. GraphML Team  
*The GraphML File Format.*  
<http://graphml.graphdrawing.org/>
10. Fritz Scholz  
*Inference for the Weibull Distribution.*  
<http://www.stat.washington.edu/fritz/DATAFILES498B2008/WeibullBounds.pdf>