



UNIVERSITÀ DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Informatica

ELABORATO FINALE

INFLUENTIAL SPREADERS IN MULTILAYER NETWORKS

Supervisore

Montresor Alberto

Laureando

Masina Gabriele

Anno accademico 2019/2020

Ringraziamenti

...thanks to...

Indice

Sommario	2
1 Introduzione	2
1.1 <i>Influential spreaders</i> in un grafo	2
1.2 <i>Influential spreaders</i> in una multilayer network	2
2 Definizioni	3
3 Algoritmi utilizzati	3
3.1 Algoritmi sul grafo aggregato	3
3.1.1 PageRank	5
3.1.2 k-core	5
3.1.3 Degree	5
3.1.4 Algoritmi applicati su ogni layer	5
3.1.5 Additive PageRank	5
3.1.6 SumCore	6
3.1.7 Algoritmi applicati sull'intera struttura	6
3.1.8 Versatility PageRank	6
3.1.9 Versatility Betweenness Centrality	6
3.1.10 Multiplex k-core	6
3.1.11 Power Community Index	6
4 Dataset	7
4.1 Estrazione reti multiplex	7
4.2 Generazione reti multilayer	7
5 Simulazioni	8
5.1 Probabilità di contagio	10
6 Risultati	10
6.1 Metodo di valutazione	10
6.2 Confronto dei risultati	11
Bibliografia	13
A Titolo primo allegato	16
A.1 Titolo	16
A.1.1 Sottotitolo	16
B Titolo secondo allegato	17
B.1 Titolo	17
B.1.1 Sottotitolo	17

Sommario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Sommario è un breve riassunto del lavoro svolto dove si descrive l'obiettivo, l'oggetto della tesi, le metodologie e le tecniche usate, i dati elaborati e la spiegazione delle conclusioni alle quali siete arrivati.

Il sommario dell'elaborato consiste al massimo di 3 pagine e deve contenere le seguenti informazioni:

- contesto e motivazioni
- breve riassunto del problema affrontato
- tecniche utilizzate e/o sviluppate
- risultati raggiunti, sottolineando il contributo personale del laureando/a

1 Introduzione

Lo studio di un processo di diffusione all'interno di una rete è molto utile per comprendere le dinamiche di diversi scenari. Tra questi, la diffusione di una notizia in una rete di contatti, di un virus in una società, una fake news in un social network e l'accumulo di un ritardo nelle stazioni di una rete di trasporti.

Di particolare interesse è l'individuazione dei cosiddetti *influential spreaders*, ovvero dei nodi che, se infettati, permettono di 'contagiare' una grossa parte della rete.

1.1 *Influential spreaders* in un grafo

Poichè i processi di simulazione della diffusione sono computazionalmente costosi, un problema di grande importanza è quello di individuare gli *influential spreaders* analizzando la topologia della rete.

Il problema è stato molto studiato nei grafi 'classici', dove i nodi e gli archi che li collegano appartengono alla stessa rete [9][3][12]. È stato osservato che alcuni algoritmi di centralità forniscono una buona indicazione di quali sono i nodi più influenti nella diffusione. Tra questi, *PageRank*, *Betweenness Centrality* e *k-core* sono algoritmi che richiedono una conoscenza globale della topologia della rete, mentre per altri come Degree Centrality o μ -PCI ogni nodo ha bisogno solo di una conoscenza locale della rete.

1.2 *Influential spreaders* in una multilayer network

In molti dei problemi reali i nodi sono collegati tramite diversi tipi di interazioni, che non possono essere rappresentate in un grafo 'classico'. Esempi di questo tipo sono la rete dei profili online collegati dalle loro interazioni in diversi social networks (follow, amicizie, condivisioni, ecc.) e la rete delle stazioni

di una città connesse da tratte di diversi mezzi di trasporto (rete ferroviaria, rete degli autobus, piste ciclabili, ecc.).

Strutture che permettono di rappresentare queste realtà sono le *multilayer network*, ossia reti in cui sono presenti differenti tipi di relazione tra i nodi.

È stata studiata l'individuazione degli *influential spreaders* in questo tipo di reti, utilizzando algoritmi di centralità definiti su di esse.

2 Definizioni

Definizione 1 (Multilayer Network). Una *multilayer network* è una coppia $(\mathcal{G}, \mathcal{E})$ dove $\mathcal{G} = \{G_i \mid i \in \{1, 2, \dots, L\}\}$ è un insieme di *grafi* detti *layers* dove un nodo può appartenere ad uno o più *layers* ed $\mathcal{E} = \{E_{ij} \subseteq V_i \times V_j \mid i, j \in \{1, 2, \dots, L\}, i \neq j\}$ è l'insieme delle *inter-connessioni* tra nodi appartenenti a layer differenti.

Definizione 2. Data una *multilayer network* $\mathcal{M} = (\mathcal{G}, \mathcal{E})$ indichiamo con $|V|$ il numero totale di nodi, ossia la cardinalità dell'insieme $V = \bigcup_i V_i$, con N la somma del numero di nodi di ogni layer, ossia $N = \sum_i |V_i|$ e con E il numero totale di archi, $E = \sum_i |E_i| + \sum_{i \neq j} |E_{ij}|$.

Definizione 3 (Multiplex Network). Una *multiplex network* è un tipo di *multilayer network* dove le uniche *inter-connessioni* ammesse sono tra un nodo e le sue controparti negli altri layer, ossia $E_{ij} \subseteq \{(v, v) \mid v \in V_i \cap V_j\}$

Definizione 4 (Grafo aggregato). Sia $\mathcal{M} = (\mathcal{G}, \mathcal{E})$ una *multilayer network*. Definiamo il *grafo aggregato* $G_{agg} = (V_{agg}, E_{agg})$ derivato da \mathcal{M} dove $V_{agg} = V$ ed $E_{agg} = \left(\bigcup_i E_i\right) \cup \left(\bigcup_{i \neq j} E_{ij}\right)$

3 Algoritmi utilizzati

Dopo una fase di ricerca, sono stati selezionati ed implementati alcuni algoritmi di centralità solitamente usati per l'individuazione degli *influential spreaders*. Tali algoritmi assegnano ad ogni nodo un valore proporzionale alla sua centralità e quindi all'influenza esercitata da tale nodo sulla rete.

Questi possono essere divisi in tre categorie:

- algoritmi applicati sul grafo aggregato;
- algoritmi applicati separatamente su ogni layer;
- algoritmi applicati sull'intera struttura *multilayer*.

3.1 Algoritmi sul grafo aggregato

Diverse misure di centralità sono utilizzate per l'individuazione degli *influential spreaders* in grafi 'classici' ad un solo layer, detti anche *monoplex*[9][3][12]. Essendo definite per grafi ad un solo layer, sono state applicate al *grafo aggregato* derivato da ogni *multilayer network*. Le misure utilizzate sono state *PageRank*[11], *k-core*[4] e *Degree*.

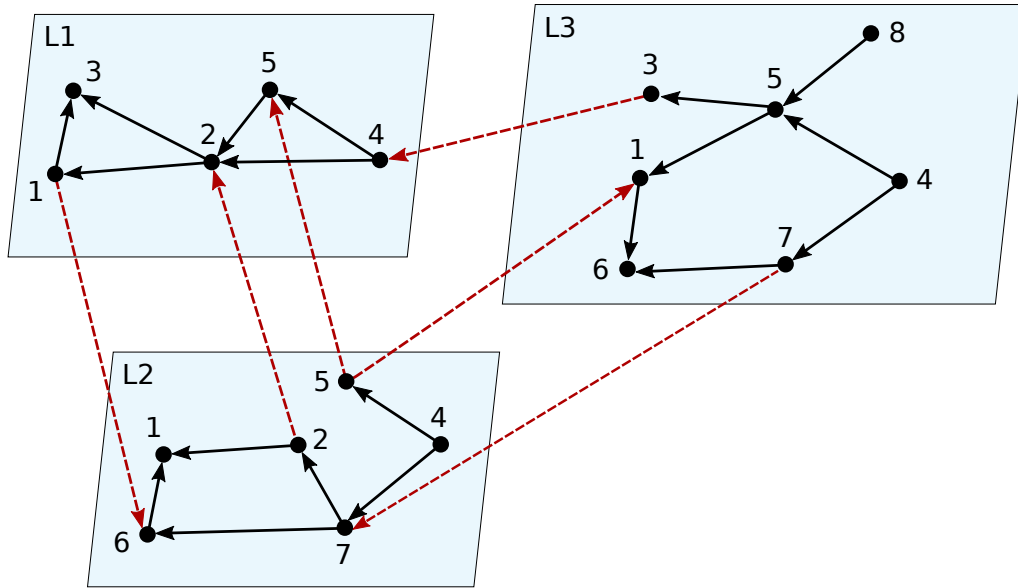


Figura 1: Rappresentazione di una *multilayer network* con tre layer: L1, L2 e L3. In rosso sono evidenziate le *inter-connessioni*, ossia gli archi che collegano nodi in layer differenti.

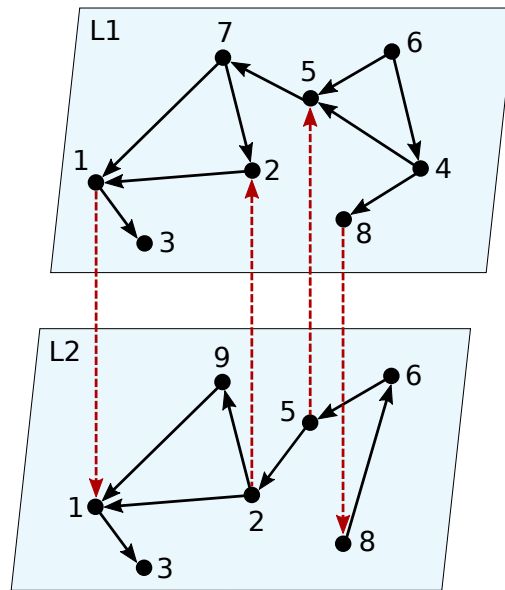


Figura 2: Rappresentazione di una *multiplex network* con due layer: L1 e L2. In questo tipo di rete le uniche *inter-connessioni* possibili sono tra un nodo e le sue controparti negli altri layer.

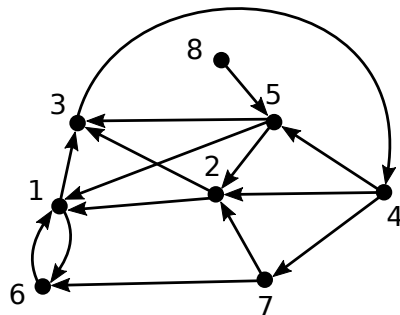


Figura 3: Grafo aggregato derivato dalla *multilayer network* rappresentata in figura 1.

3.1.1 PageRank

Definizione 5 (PageRank). Dato un grafo $G = (V, E)$, il valore di PageRank di un nodo v è definito come

$$PR(v) = \alpha \sum_{w:v \in G.adj(w)} \frac{PR(w)}{G.outdegree(w)} + (1 - \alpha) \frac{1}{|V|} \quad (3.1)$$

Questa misura di centralità fu originariamente definita per misurare l'importanza di una pagina web. Secondo questo criterio, una pagina è tanto più importante quante più pagine importanti hanno un link verso di essa. Analogamente può essere utilizzata per misurare più genericamente l'influenza di un nodo in un grafo, ed è stata applicata sia al grafo originale che a quello trasposto.

Questa definizione ricorsiva è stata così calcolata in modo iterativo:

$$PR_0(v) = \frac{1}{n}, \quad PR_i(v) = \alpha \sum_{w:v \in G.adj(w)} \frac{PR_{i-1}(w)}{G.outdegree(w)} + (1 - \alpha) \frac{1}{|V|} \quad (3.2)$$

finchè $\|PR_i - PR_{i-1}\|_1 < \epsilon$. Secondo [11] il numero di iterazioni è proporzionale a $\log |V|$. Dunque, la complessità dell'algoritmo è $O((|V| + |E|) \log |V|)$.

3.1.2 k-core

Definizione 6 (k-core). Dato un grafo $G = (V, E)$, un k -core di G è un sottografo $G' = (V', E')$ tale che

$$G'.indegree(v) \geq k \quad \forall v \in V' \quad (3.3)$$

Un nodo appartenente ad un k -core con k elevato è considerato un nodo centrale nella rete, e quindi un potenziale *influential spreader*. Per il calcolo di questa misura è stato utilizzato l'algoritmo definito in [4] con complessità $O(|V| + |E|)$. Questo algoritmo è stato applicato sia al grafo originale che a quello trasposto.

3.1.3 Degree

Questa misura di centralità assegna ad ogni nodo un valore uguale al suo *outdegree*, per cui un nodo con tanti vicini si assume possa avere una certa rilevanza in un processo di diffusione. Può essere calcolata con complessità $O(|V|)$.

3.1.4 Algoritmi applicati su ogni layer

Alcune delle misure sono state estese alle reti multilayer calcolando la centralità dei nodi in ogni layer separatamente e poi sommando i punteggi ottenuti in ogni layer. In questa categoria rientrano *Additive Pagerank*[7] e *SumCore*[2]

3.1.5 Additive PageRank

Definizione 7 (Additive PageRank). Data una *multilayer network* $\mathcal{M} = (\mathcal{G}, \mathcal{E})$ ed un ordinamento dei layer (X_1, \dots, X_L) , l'*Additive PageRank* di un nodo $v \in V$ è il valore $addPR(v) = addPR_L(v)$, dove:

$$\begin{aligned} addPR_1(v) &= PR(v) \text{ in } X_1 \\ addPR_l(v) &= \alpha \sum_{w:v \in X_l.adj(w)} \frac{addPR_l(w)}{X_l.outdegree(w)} + (1 - \alpha) \frac{addPR_{l-1}(v)}{|V| \langle addPR_{l-1} \rangle} \end{aligned} \quad (3.4)$$

Questa estensione dell'algoritmo di PageRank alle reti Multilayer richiede un ordinamento dei layer. Come fatto in [2], i layer sono stati ordinati per valore dell'autovalore di modulo massimo della rispettiva matrice di adiacenza. Infatti, autovalore più grande significa maggiore capacità di diffusione[15].

Come algoritmo, è stato utilizzato un approccio analogo a quello della versione classica di PageRank descritto in 3.1.1. Poichè è necessario calcolare il PageRank su ogni layer, il costo dell'algoritmo è $O(\sum_i ((|V_i| + |E_i|) \log(|V_i|)))$, assumendo che i layer siano già ordinati.

Questa misura è stata calcolata sia sulla rete originale che su quella trasposta.

3.1.6 SumCore

Questa misura è stata ottenuta calcolando il valore di k -core di ogni nodo in ogni layer, quindi sommando i valori ottenuti da un nodo in tutti i layer. Pertanto la complessità del calcolo è pari a $O(\sum_i (|V_i| + |E_i|))$.

Questa misura è stata calcolata sia sulla rete originale che su quella trasposta.

3.1.7 Algoritmi applicati sull'intera struttura

Diverse misure definite originariamente per grafi *monoplex* sono state estese a reti *multilayer*. Queste, al contrario di quelle presentate nei punti precedenti, tengono conto della struttura a più livelli della rete ed in particolare delle *inter-conessioni*. Quelle implementate sono *versatility PageRank*[6], *versatility Betweenness Centrality*[6] [14], *multiplex k-core*[1], *minimal-layers PCI*[2], *layer-agnostic PCI*[2], *all-layers PCI*[2], *layer-symmetric PCI*[2].

3.1.8 Versatility PageRank

Definizione 8 (Versatility PageRank). Data una *multilayer network* $\mathcal{M} = (\mathcal{G}, \mathcal{E})$, si definisce *Versatility PageRank* di un nodo $v \in V_l$ in un layer l il valore:

$$PR(v, l) = \alpha \left(\sum_{w: v \in G_l, \text{adj}(w)} \frac{PR(w, l)}{\mathcal{M.outdegree}(w, l)} + \sum_{j \neq l} \sum_{w: (w, v) \in E_{jl}} \frac{PR(w, j)}{\mathcal{M.outdegree}(w, j)} \right) + (1 - \alpha) \frac{1}{N} \quad (3.5)$$

dove $\mathcal{M.outdegree}(w, l)$ è la somma del numero di archi uscenti da w nel layer l e del numero di *inter-conessioni* uscenti da w nel layer l .

La definizione originale in [6] utilizzava un tensore 4-dimensionale per rappresentare la rete. Qui è stata generalizzata, anche prevedendo che un nodo possa non comparire in tutti i layer.

Questa definizione è stata calcolata in modo analogo alla definizione 3.1.1 nella pagina precedente, dunque la complessità è $O((N + E) \log(N))$. Questa misura è stata applicata sia alla rete originale che a quella trasposta.

3.1.9 Versatility Betweenness Centrality

Definizione 9 (Versatility Betweenness Centrality). Data una *multilayer network* $\mathcal{M} = (\mathcal{G}, \mathcal{E})$, la *Versatility Betweenness Centrality* di un nodo v è il valore

$$BC(v) = \sum_{s, t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (3.6)$$

dove σ_{st} è il numero di percorsi minimi tra il nodo s ed il nodo t indipendentemente dal layer, e $\sigma_{st}(v)$ è il numero di questi che passa per il nodo v in qualche layer.

Questa misura è stata calcolata con l'algoritmo definito in [14], con complessità $O(|V|(N + E))$.

3.1.10 Multiplex k-core

Definizione 10 (Multiplex k-core). Data una *multilayer network* $\mathcal{M} = (\mathcal{G}, \mathcal{E})$, il k -core è il più grande sottografo per cui ogni nodo è raggiunto in ogni layer da almeno k archi o *inter-conessioni*.

Questa misura è stata calcolata estendendo l'algoritmo definito in [4] per adattarlo a questa definizione, ottenendo una complessità di $O(N + E)$.

3.1.11 Power Community Index

Definizione 11. Data una *multilayer network* $\mathcal{M} = (\mathcal{G}, \mathcal{E})$, l'indice $mlPCI_n$ di un nodo v in un layer l è il massimo numero k tale che esistono almeno k vicini di v nel layer l in \mathcal{M} con numero di vicini in almeno n layer maggiore o uguale a k .

Da questa definizione si ricavano le seguenti:

Definizione 12 (minimal-layers PCI). L'indice $mlPCI$ di un nodo v in un layer l è definito come

$$mlPCI(v, l) = \sum_{n=1}^L mlPCI_n(v, l) \quad (3.7)$$

Definizione 13 (all-layers PCI). L'indice $alPCI$ di un nodo v in un layer l è definito come

$$alPCI(v, l) = mlPCI_L(v, l) \quad (3.8)$$

Definizione 14 (layer-symmetric PCI). L'indice $lsPCI$ di un nodo v in un layer l è il massimo numero k tale che esistono almeno k vicini di v nel layer l in \mathcal{M} con numero di vicini in almeno k layer maggiore o uguale a k .

Definizione 15 (layer-agnostic PCI). L'indice $laPCI$ di un nodo v in un layer l è il massimo numero k tale che esistono almeno k vicini di v nel layer l in \mathcal{M} con numero di vicini maggiore o uguale a k .

Il calcolo di queste definizioni, è stato fatto utilizzando un algoritmo di complessità $O(NLt \log t)$, dove $t = \max\{\mathcal{M}.outdegree(v) | v \in V\}$

4 Dataset

Queste misure sono state testate su diverse reti con caratteristiche diverse, seguendo una metodologia simile a [2]:

- Multiplex Network estratte da dataset biologici e sociali;
- Multilayer Network generate a partire da reti di interazioni in Internet.

4.1 Estrazione reti multiplex

Le reti indicate in tabella 4.1 nella pagina seguente sono state estratte a partire da reti biologiche e sociali fornite da [5]. L'obiettivo era quello di ottenere un dataset simile a quello utilizzato da [2], dove per ogni rete sono indicati i numeri N , E ed $|L|$, rispettivamente di nodi, archi e layer estratti.

Per ogni rete, è stato individuato un sottoinsieme L di $|L|$ layer in cui il numero di nodi con una controparte in ogni layer fosse almeno N . Per tutte le reti, è stato trovato un solo sottoinsieme che rispettasse questo criterio.

Dopo aver costruito una sotto-rete formata solo dai layer in L , sono stati rimossi i nodi non presenti in tutti i layer e gli archi da e verso essi. Dopodichè sono stati rimossi tutti i nodi da cui non arrivava nè partiva più nessun arco. Questa procedura è stata applicata ricorsivamente alla sotto-rete ottenuta, finchè non è stato più possibile rimuovere alcun nodo.

I risultati ottenuti sono stati molto simili a quelli voluti, come si può vedere in tabella 4.1 nella pagina successiva.

In queste reti sono state aggiunte *inter-conessioni* tra le controparti di ogni nodo in ogni layer.

4.2 Generazione reti multilayer

Sono state generate delle *multilayer network* partendo da grafi di interazioni su social network e su architetture peer-to-peer, in particolare quelli indicati in tabella 4.2 a pagina 10, che si possono trovare in [10].

Seguendo l'approccio di [2], sono stati generati due tipi di reti:

- Similar Layers Network(SLN), i cui layer sono i grafi [3-6] della tabella 4.2 a pagina 10. In queste reti i layer hanno tutti un numero simile di nodi e archi;
- Different Layers Networks(DLN), i cui layer sono i grafi [1-3] della tabella 4.2 a pagina 10. In queste reti i layer hanno numero di nodi e archi molto differenti.

Tabella 4.1: Reti multiplex estratte

Nome	Nodi			Archi			Layers	
	iniziali	obiettivo	estratti	iniziali	obiettivo	estratti	iniziali	estratti
Drosophila	8215	1364	1375	43 366	7267	7438	7	2
Homo	18 222	3859	3878	170 899	77 483	78 804	7	3
MA2013	88 804	4370	4377	210 250	33 411	34 589	3	3
NYCM2014	102 439	4150	4262	353 495	45 334	47 840	3	3
SacchCere	6570	3096	3121	28 275	185 849	188 182	7	5
SacchPomb	4092	875	878	63 676	18 214	18 308	7	3

Generazione delle *inter-conessioni* Per prima cosa, sono state aggiunte *inter-conessioni* tra le controparti di ogni nodo in tutti i layer in cui esso è presente.

Quindi, sono state generate casualmente altre *inter-conessioni* impostando tre parametri:

1. il numero di *inter-conessioni* uscenti da un nodo in un determinato layer;
2. la frequenza con cui un layer viene scelto come layer di destinazione di una connessione;
3. la frequenza con cui un nodo in un certo layer viene scelto come nodo di destinazione.

Ognuno di questi parametri è stato generato secondo una distribuzione che segue la legge di Zipf [16], per cui la frequenza di un certo valore è inversamente proporzionale al valore stesso.

Tale distribuzione genera numeri tra 1 ed un valore massimo m con diverse probabilità, secondo il valore di una variabile $s \in [0, 1]$ che regola il grado di ‘asimmetria’ dei valori generati: per $s = 0$ la probabilità di ogni valore è la stessa, mentre per $s = 1$, si ha $\mathcal{P}(1) = 2 \cdot \mathcal{P}(2) = 3 \cdot \mathcal{P}(3) = \dots = m \cdot \mathcal{P}(m)$.

Le variabili che regolano i tre parametri sono chiamate rispettivamente s_{degree} , s_{layer} , s_{node} . Per ognuna sono stati sperimentati i valori 0.3 e 0.8. Per quanto riguarda il numero di *inter-conessioni* di un nodo in un determinato layer il valore massimo è stato impostato a $d \cdot \log_2 \sum_i V_i$, con $d = 2$. Per gli altri parametri, invece, tutti i layer devono essere selezionabili così come tutti i nodi all’interno di un layer.

La probabilità che un nodo abbia k *inter-conessioni* è $\mathcal{P}_{degree}(k)$. Per la scelta del layer di destinazione si è operato come segue: è stata generata una permutazione casuale dei layer; definita $pos(l)$ la posizione del layer l nella permutazione, la probabilità che questo venga scelto come layer di destinazione è $\mathcal{P}_{layer}(pos(l))$. In modo analogo sono stati permutati i nodi all’interno di ogni layer per determinarne la probabilità di essere scelti come destinazione.

Le reti così generate sono denominate con $SLN_{s_{degree}, s_{layer}, s_{node}}$ oppure $DLN_{s_{degree}, s_{layer}, s_{node}}$.

L’algoritmo 1 nella pagina successiva mostra come sono state generate le reti.

5 Simulazioni

Per valutare se un algoritmo possa essere utilizzato per riconoscere gli *influential spreaders*, sono stati confrontati i valori di centralità che esso assegna ai vari nodi con la frazione di rete infettata da essi simulando un processo di diffusione, detto *spreading process*.

Lo *spreading process* utilizzato segue il modell *SIR*, in cui un nodo può essere in 3 stati:

- *Susceptible* (S) - ossia vulnerabile ad essere ‘infettato’;
- *Infectious* (I) - ossia ‘infetto’. Un nodo in questo stato può contagiare i suoi vicini con una certa probabilità;
- *Recovered* (R) - ossia guarito. Una volta in questo stato, un nodo non può più essere infettato.

Algoritmo 1: Generazione multilayer networks

```
Input : GRAPH layers[] // grafi usati come layers
        int perm_l[] // posizione di ogni layer nella permutazione
        int perm_n[][] // posizione di un nodo nella permutazione di ogni layer
        int d
        float s_degree
        float s_layer
        float s_node

1 int total_nodes = 0
2 for i = 0 to layers.size() do
3   total_nodes = total_nodes + layers[i].nodes().size()

4 int max_interconnections = d ·  $\lfloor \log_2(\text{total\_nodes}) \rfloor$ 
   // I due parametri della classe ZIPFGENERATOR regolano rispettivamente
   // il massimo della distribuzione e il grado di asimmetria
5 ZIPFGENERATOR degree_generator(max_interconnections, s_degree)
6 ZIPFGENERATOR layer_generator(layers.size(), s_layer)
7 ZIPFGENERATOR node_generators[0...layers.size() - 1]
8 for l = 0 to layers.size() - 1 do
9   node_generators[l] = ZIPFGENERATOR(layers[l].size(), s_degree)

10 MULTILAYERNETWORK m
11 for l = 0 to layers.size() - 1 do
12   foreach n ∈ layers[l] do
13     // intra-conessioni
14     for v ∈ layers[l].adj(n) do
15       m.add_edge(n, l, v, l)
16     // inter-conessioni tra un nodo e le sue controparti negli altri layer
17     for j = 0 to layers.size() - 1 do
18       if j ≠ l and n ∈ layers[j].nodes() then
19         m.add_edge(n, l, n, j)
20     // Generazione inter-conessioni casuali
21     int degree = degree_generator.next()
22     for i = 1 to degree do
23       bool added = False
24       while not added do
25         int l_dest = l_index[layer_generator.next()]
26         while l_dest == l do
27           l_dest = l_index[layer_generator.next()]
28         int n_dest = n_index[l_dest][node_generators[l_dest].next()]
29         if not m.has_edge(n, l, n_dest, l_dest) then
30           added = True
31           m.add_edge(n, l, n_dest, l_dest)

29 return m
```

Tabella 4.2: Dataset per la generazione di reti multilayer

Numero	Nome	Nodi	Archi
1.	wiki-Vote	7115	103 689
2.	cit-HepTh	27 770	352 807
3.	p2p-Gnutella04	10 876	39 994
4.	p2p-Gnutella05	8846	31 839
5.	p2p-Gnutella06	8717	31 525
6.	p2p-Gnutella08	6301	20 777

5.1 Probabilità di contagio

Un valore da impostare in uno *spreading process* è la probabilità λ che un nodo nello stato I ha di contagiare i propri vicini nello stato S , detta *epidemic probability*. La scelta del valore di λ è molto importante per riconoscere gli *influential spreaders*, in quanto con un valore troppo alto si osserverebbe la diffusione dell'infezione in tutta la rete indipendentemente dal nodo inizialmente infetto, mentre con un valore troppo basso l'epidemia non riuscirebbe ad espandersi nemmeno dai nodi più influenti.

In un grafo *monoplex* G il valore di *critical epidemic point*

$$\lambda_c = \frac{\langle k \rangle}{\langle k^2 \rangle} \quad (5.1)$$

dove k è l'*outdegree* di un nodo, rappresenta un'approssimazione della soglia per cui, indipendentemente dal nodo da cui parte l'epidemia, se $\lambda > \lambda_c$ essa riesce a diffondersi in gran parte della rete, mentre se $\lambda < \lambda_c$ essa rimane confinata in una piccola parte della rete[13].

Seguendo quanto fatto in [2], la *epidemic probability* λ_{ii} da un nodo nel layer i ad uno dello stesso layer è stata impostata al *critical epidemic point* del layer G_i , mentre λ_{ij} tra nodi in layer diversi è stata impostata al *critical epidemic point* del grafo aggregato.

Questi valori rispecchiano l'intuizione secondo cui il contagio si trasmette più facilmente tra nodi sullo stesso layer. Questo succede per esempio nel caso dei social network, in cui diverse piattaforme tendono a mostrare in maniera minore contenuti con collegamenti a piattaforme concorrenti, e nel caso dei ritardi accumulati da una stazione in una rete di trasporti, che possono trasmettersi facilmente alle stazioni vicine nella stessa rete (es. rete ferroviaria), ma anche, indirettamente e presumibilmente in maniera minore, ad altre reti di cui la stazione fa parte.

6 Risultati

6.1 Metodo di valutazione

Per ogni rete è stato osservato il numero di nodi infettati da ogni nodo facendo una media di 100 simulazioni e ottenendo così un vettore s di $|V|$ elementi, dove s_i rappresenta il numero di nodi infettati in media in uno *spreading process* in cui i nodi inizialmente infetti sono le controparti del nodo i nei layer in cui esso è presente.

È stato poi calcolato per ogni algoritmo un vettore a dove a_i è il punteggio assegnato dall'algoritmo al nodo i .

Per confrontare i vettori così ottenuti ed assegnare un punteggio ad ogni algoritmo che ne valutasse la capacità di riconoscere gli *influential spreaders* è stato utilizzato coefficiente *Kendall's τ* [8], così definito:

$$\tau = \frac{n_c - n_d}{n(n-1)} \quad (6.1)$$

dove:

- n è il numero di elementi dei due vettori, ossia il numero di nodi —V—;
- n_c è il numero di coppie concordanti nei due vettori. Una coppia di elementi indici i, j si dice concordante se $a_i > a_j$ e $s_i > s_j$ oppure se $a_i < a_j$ e $s_i < s_j$;
- n_d è il numero di coppie discordanti nei due vettori. Una coppia di elementi indici i, j si dice discordante se $a_i > a_j$ e $s_i < s_j$ oppure se $a_i < a_j$ e $s_i > s_j$;

Se $a_i = a_j$ oppure $s_i = s_j$, la coppia i, j non viene contata.

Il risultato della formula è compreso tra -1.0 e 1.0

6.2 Confronto dei risultati

Le tabelle 6.1, 6.2 e 6.3 mostrano i punteggi ottenuti da ogni algoritmo nelle reti DLN, SLN e Multiplex rispettivamente. In ogni tabella sono evidenziati per ogni rete i punteggi ottenuti dai 3 algoritmi migliori.

Si può notare che *mlPCI* e *laPCI* sono i due algoritmi che ottengono punteggi buoni su tutte le reti utilizzate, e possono quindi essere considerati un buon metodo per indentificare gli *influential spreaders*.

Tabella 6.1: Kendall's tau di diversi algoritmi in reti DLN

	DLN _{0.3, 0.3, 0.3}	DLN _{0.3, 0.3, 0.8}	DLN _{0.3, 0.8, 0.3}	DLN _{0.3, 0.8, 0.8}	DLN _{0.8, 0.3, 0.3}	DLN _{0.8, 0.3, 0.8}	DLN _{0.8, 0.8, 0.3}	DLN _{0.8, 0.8, 0.8}
addPR	0.4885	0.4844	0.4888	0.4860	0.4872	0.4874	0.4853	0.4848
aggCore	0.5575	0.5376	0.5552	0.5379	0.5596	0.5391	0.5578	0.5416
aggDeg	0.6620	0.6402	0.6607	0.6401	0.6274	0.6201	0.6297	0.6179
aggPR	0.5316	0.4958	0.5296	0.4972	0.5214	0.4879	0.5195	0.4869
alPCI	0.6945	0.7036	0.7043	0.7109	0.6565	0.6694	0.6776	0.6800
laPCI	0.7145	0.7157	0.7113	0.7158	0.6759	0.6910	0.6783	0.6868
lsPCI	0.6313	0.6285	0.6354	0.6335	0.6840	0.6848	0.6879	0.6895
mlPCI	0.7344	0.7406	0.7305	0.7427	0.6980	0.7114	0.7027	0.7095
multiCore	0.3551	0.3574	0.3554	0.3576	0.3551	0.3549	0.3553	0.3550
sumCore	0.5225	0.5350	0.5259	0.5346	0.5279	0.5350	0.5259	0.5334
verBC	0.6586	0.5605	0.6610	0.5639	0.6275	0.5404	0.6309	0.5441
verPR	0.5275	0.5141	0.5261	0.5136	0.5195	0.5078	0.5162	0.5064

Algoritmi applicati al grafo trasposto Le diverse versioni degli algoritmi di *PageRank* e *k-core* sono state applicate anche al grafo trasposto.

Nel caso di *PageRank*, se nella definizione sul grafo originale un nodo è tanto più importante quanto più importanti sono i nodi che hanno un arco verso di esso, applicando l'algoritmo sul grafo trasposto si ottiene una definizione per cui un nodo è tanto più importante quanto più importanti sono i nodi verso cui ha un arco.

Per quanto riguarda *k-core*, invece, nella definizione sul grafo originale un *k-core* di un grafo è un sottografo massimale in cui ogni nodo ha almeno k archi entranti, applicando l'algoritmo sul grafo trasposto si ottiene una definizione analoga, in cui invece degli archi entranti si considerano gli archi uscenti.

Tabella 6.2: Kendall's tau di diversi algoritmi in reti SLN

	SLN _{0.3, 0.3, 0.3}	SLN _{0.3, 0.3, 0.8}	SLN _{0.3, 0.8, 0.3}	SLN _{0.3, 0.8, 0.8}	SLN _{0.8, 0.3, 0.3}	SLN _{0.8, 0.3, 0.8}	SLN _{0.8, 0.8, 0.3}	SLN _{0.8, 0.8, 0.8}
addPR	0.4225	0.4293	0.4216	0.4295	0.4198	0.4309	0.4178	0.4287
aggCore	0.4066	0.3684	0.4158	0.3731	0.4010	0.3795	0.4117	0.3841
aggDeg	0.6073	0.5520	0.6009	0.5460	0.6147	0.5490	0.6134	0.5412
aggPR	0.4275	0.3619	0.4313	0.3695	0.4229	0.3713	0.4294	0.3779
alPCI	0.6550	0.6152	0.6379	0.5971	0.6675	0.6300	0.6555	0.6203
laPCI	0.6243	0.5761	0.6195	0.5687	0.6685	0.6208	0.6662	0.6092
lsPCI	0.5901	0.5681	0.5728	0.5479	0.5962	0.5678	0.5897	0.5584
mlPCI	0.6265	0.5810	0.6222	0.5771	0.6660	0.6169	0.6627	0.6057
multiCore	0.3481	0.3560	0.3498	0.3576	0.3506	0.3535	0.3472	0.3513
sumCore	0.4209	0.4288	0.4211	0.4273	0.4164	0.4291	0.4171	0.4267
verBC	0.5589	0.4528	0.5480	0.4506	0.5868	0.4842	0.5739	0.4796
verPR	0.3650	0.3479	0.3697	0.3579	0.3200	0.3297	0.3317	0.3374

Tabella 6.3: Kendall's tau di diversi algoritmi in reti multiplex

	Drosophila	Homo	MA2013	NYCM2014	SacchCere	SacchPomb
addPR	0.0398	0.3424	0.0059	-0.1313	0.3300	0.3410
aggCore	0.0646	0.4132	0.0572	-0.0453	0.4384	0.1046
aggDeg	0.7355	0.7096	0.5711	0.6150	0.6886	0.7656
aggPR	0.0417	0.3857	0.0164	-0.0771	0.3944	0.2562
alPCI	0.3682	0.1588	0.0493	0.1124	0.0455	0.4502
laPCI	0.6040	0.6859	0.5534	0.6178	0.6980	0.6853
lsPCI	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
mlPCI	0.6947	0.7191	0.5532	0.6028	0.7073	0.7729
multiCore	0.0497	0.1962	0.0428	0.0230	0.1925	-0.0170
sumCore	0.0658	0.4575	0.0616	-0.0487	0.4546	0.1000
verBC	0.5243	0.5653	0.1726	0.2045	0.5666	0.6550
verPR	-0.3505	-0.0606	-0.2707	-0.4107	0.0004	-0.4201

Per entrambi gli algoritmi si è pensato che le definizioni di centralità ottenute applicando l'algoritmo sul grafo trasposto potessero individuare meglio gli *influential spreaders*. Le prestazioni di questi algoritmi, indicati con l'indice T , sono mostrate nelle tabelle 6.4, 6.5 e 6.6.

In generale, si può osservare come nella maggior parte dei casi questi ottengano risultati migliori se confrontati con la rispettiva versione applicata al grafo originale.

Tabella 6.4: Kendall's tau di algoritmi applicati al grafo trasposto in reti DLN

	DLN _{0.3,} 0.3, 0.3	DLN _{0.3,} 0.3, 0.8	DLN _{0.3,} 0.8, 0.3	DLN _{0.3,} 0.8, 0.8	DLN _{0.8,} 0.3, 0.3	DLN _{0.8,} 0.3, 0.8	DLN _{0.8,} 0.8, 0.3	DLN _{0.8,} 0.8, 0.8
addPR T	0.3649	0.4137	0.3601	0.4134	0.3744	0.4136	0.3755	0.4110
aggCore T	0.6063	0.5604	0.5946	0.5740	0.5718	0.5698	0.5823	0.5809
aggPR T	0.6161	0.5469	0.6115	0.5461	0.5484	0.4965	0.5496	0.4930
multiCore T	0.3646	0.3635	0.3647	0.3632	0.3648	0.3633	0.3649	0.3644
sumCore T	0.4377	0.4813	0.4340	0.4824	0.4468	0.4816	0.4474	0.4804
verPR T	0.6985	0.6547	0.6934	0.6535	0.6645	0.6554	0.6647	0.6364

Tabella 6.5: Kendall's tau di algoritmi applicati al grafo trasposto in reti SLN

	SLN _{0.3,} 0.3, 0.3	SLN _{0.3,} 0.3, 0.8	SLN _{0.3,} 0.8, 0.3	SLN _{0.3,} 0.8, 0.8	SLN _{0.8,} 0.3, 0.3	SLN _{0.8,} 0.3, 0.8	SLN _{0.8,} 0.8, 0.3	SLN _{0.8,} 0.8, 0.8
addPR T	0.4674	0.4831	0.4714	0.4829	0.4675	0.4824	0.4676	0.4849
aggCore T	0.4608	0.4225	0.4601	0.4300	0.5110	0.4362	0.4811	0.4663
aggPR T	0.5900	0.5700	0.5822	0.5625	0.5814	0.5413	0.5815	0.5362
multiCore T	0.4138	0.3830	0.4135	0.4085	0.4137	0.4084	0.4139	0.4068
sumCore T	0.6339	0.6559	0.6382	0.6565	0.6296	0.6545	0.6316	0.6566
verPR T	0.6210	0.6245	0.6252	0.6251	0.6323	0.6240	0.6386	0.6207

Tabella 6.6: Kendall's tau di algoritmi applicati al grafo trasposto in reti multiplex

	Drosophila	Homo	MA2013	NYCM2014	SacchCere	SacchPomb
addPR T	0.6497	0.5770	0.3193	0.4235	0.5356	0.6701
aggCore T	0.5686	0.7138	0.4756	0.5475	0.7073	0.7640
aggPR T	0.6441	0.7033	0.3156	0.4263	0.7202	0.7435
multiCore T	0.0763	0.2165	0.1491	0.0127	0.2569	0.2110
sumCore T	0.4377	0.6689	0.4791	0.5380	0.6899	0.7464
verPR T	0.5280	0.5797	0.2286	0.3878	0.6370	0.6799

Bibliografía

- [1] N. Azimi-Tafreshi, J. Gómez-Gardeñes, and S. N. Dorogovtsev. k -core percolation on multiplex networks. *Phys. Rev. E*, 90:032816, Sep 2014.
- [2] P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas. Identifying influential spreaders in complex multilayer networks: A centrality perspective. *IEEE Transactions on Network Science and Engineering*, 6(1):31–45, Jan 2019.
- [3] P. Basaras, D. Katsaros, and L. Tassiulas. Detecting influential spreaders in complex, dynamic networks. *Computer*, 46(4):24–29, April 2013.
- [4] Vladimir Batagelj and Matjaz Zaversnik. An $o(m)$ algorithm for cores decomposition of networks. *CoRR*, cs.DS/0310049, 2003.
- [5] FBK CoMuNe Lab. Datasets released for reproducibility. <https://comunelab.fbk.eu/data.php>.
- [6] Manlio De Domenico, Albert Solé-Ribalta, Elisa Omodei, Sergio Gómez, and Alex Arenas. Ranking in interconnected multilayer networks reveals versatile nodes. *Nature Communications*, 6(1):6868, Apr 2015.
- [7] Arda Halu, Raúl J. Mondragón, Pietro Panzarasa, and Ginestra Bianconi. Multiplex pagerank. *PloS one*, 8(10):e78293–e78293, Oct 2013.
- [8] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 06 1938.
- [9] Maksim Kitsak, Lazaros K. Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H. Eugene Stanley, and Hernán A. Makse. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893, Nov 2010.
- [10] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- [11] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.
- [12] Sen Pei, Lev Muchnik, José S. Andrade Jr., Zhiming Zheng, and Hernán A. Makse. Searching for superspreaders of information in real-world social media. *Scientific Reports*, 4(1):5547, Jul 2014.
- [13] Anna Saumell-Mendiola, M. Ángeles Serrano, and Marián Boguñá. Epidemic spreading on interconnected networks. *Phys. Rev. E*, 86:026106, Aug 2012.
- [14] Albert Solé-Ribalta, Manlio De Domenico, Sergio Gómez, and Alex Arenas. Centrality rankings in multiplex networks. In *Proceedings of the 2014 ACM Conference on Web Science, WebSci '14*, page 149–155, New York, NY, USA, 2014. Association for Computing Machinery.
- [15] Yang Wang, D. Chakrabarti, Chenxi Wang, and C. Faloutsos. Epidemic spreading in real networks: an eigenvalue viewpoint. In *22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.*, pages 25–34, Oct 2003.

- [16] George Kingsley Zipf. *Human behavior and the principle of least effort*. Human behavior and the principle of least effort. Addison-Wesley Press, Oxford, England, 1949.

Allegato A Titolo primo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Allegato B Titolo secondo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.