



# **Università Degli Studi Di Roma Tor Vergata**

Macroarea Di Scienze Matematiche Fisiche E Naturali  
Corso Di Laurea Triennale In Matematica  
Tesi Di Laurea Triennale In Matematica

---

## ***Diffusione Dei Virus Su Reti Informatiche***

---

Studente:  
Matteo Vicari

Relatore:  
Prof. G. Scalia Tomba

*Anno Accademico 2021/2022*

|          |                                                   |           |
|----------|---------------------------------------------------|-----------|
| <b>1</b> | <b>Introduzione</b>                               | <b>1</b>  |
| <b>2</b> | <b>Modelli stocastici e deterministici</b>        | <b>2</b>  |
| <b>3</b> | <b>Modello deterministico classico</b>            | <b>4</b>  |
| 3.1      | Classificazione della popolazione . . . . .       | 4         |
| 3.2      | Parametri . . . . .                               | 5         |
| 3.3      | SI . . . . .                                      | 5         |
| 3.4      | SIS . . . . .                                     | 5         |
| 3.4.1    | Formulazione deterministica e soluzione . . . . . | 6         |
| 3.4.2    | Relazione tra $\beta$ e $N$ . . . . .             | 7         |
| <b>4</b> | <b>Reti</b>                                       | <b>8</b>  |
| 4.1      | Definizioni generali . . . . .                    | 8         |
| 4.2      | Metrica delle reti . . . . .                      | 9         |
| 4.2.1    | Percorso più corto e diametro . . . . .           | 9         |
| 4.2.2    | Grado . . . . .                                   | 9         |
| 4.2.3    | Correlazione tra gradi . . . . .                  | 9         |
| 4.2.4    | Centralità . . . . .                              | 9         |
| 4.3      | Classi di reti e modelli . . . . .                | 9         |
| 4.3.1    | Networks omogenei . . . . .                       | 10        |
| 4.3.2    | Small-World networks . . . . .                    | 10        |
| 4.3.3    | Heavy-tailed networks . . . . .                   | 10        |
| 4.3.4    | Reti statiche e dinamiche . . . . .               | 10        |
| <b>5</b> | <b>Processo Markoviano</b>                        | <b>12</b> |
| 5.1      | Catene di Markov . . . . .                        | 12        |
| 5.2      | Matrice di transizione . . . . .                  | 13        |
| 5.3      | Catene di Markov aperiodiche . . . . .            | 13        |
| 5.4      | Catene di Markov irriducibili . . . . .           | 14        |
| 5.5      | Stati ricorrenti positivi . . . . .               | 14        |
| 5.6      | Distribuzioni stazionarie . . . . .               | 14        |

|          |                                                                  |           |
|----------|------------------------------------------------------------------|-----------|
| <b>6</b> | <b>Modello stocastico SIS semplice</b>                           | <b>15</b> |
| 6.1      | Legge dei grandi numeri . . . . .                                | 15        |
| 6.2      | Tempo all'estinzione . . . . .                                   | 16        |
| <b>7</b> | <b>Mean Field Theory per Epidemie su Grafi</b>                   | <b>17</b> |
| 7.1      | IBMF . . . . .                                                   | 17        |
| 7.2      | Approccio IBMF sul modello SIS . . . . .                         | 18        |
| <b>8</b> | <b>Simulazione di epidemie SIS deterministiche e stocastiche</b> | <b>22</b> |
| 8.1      | Modello SIS deterministico . . . . .                             | 23        |
| 8.2      | Epidemia su Network Omogeneo . . . . .                           | 24        |
| 8.3      | Epidemia su Network Small-World . . . . .                        | 26        |
| 8.4      | Epidemia su Network Scale-Free . . . . .                         | 28        |
| <b>9</b> | <b>Confronto risultati</b>                                       | <b>31</b> |
| 9.1      | Andamenti . . . . .                                              | 31        |
| 9.2      | Risoluzione di (7.4) nel caso stazionario . . . . .              | 33        |
| <b>A</b> | <b>Tempo all'estinzione epidemica</b>                            | <b>38</b> |
| <b>B</b> | <b>Metodo del Gradient Descent</b>                               | <b>39</b> |

# CAPITOLO 1

---

## Introduzione

---

I virus informatici rimangono una minaccia significativa per le reti e i sistemi odierni.[2] I meccanismi di difesa esistenti si concentrano sulla scansione in locale delle firme dei virus. Questi meccanismi possono rilevare e prevenire la diffusione di noti malware ma si dimostrano di scarso aiuto nella progettazione di difese ottimali a livello globale. La recente proliferazione di malware in grado di diffondersi autonomamente, senza il diretto intervento di un utente, rende la questione ancor più delicata ma fornisce un punto di contatto con lo studio di epidemie biologiche. Questi ultimi tipi di malware (comunemente noti col termine *worms*) hanno infatti caratteristiche molto simili ai virus biologici e ciò permetterà di poter applicare modelli già noti di epidemie naturali su epidemie digitali. Con l'eccezione di alcuni studi di modellizzazione specializzati, molto rimane ancora sconosciuto sulle caratteristiche di propagazione di script malevoli e i fattori che li influenzano. Le domande determinanti in questi casi sono: il malware riuscirà ad invadere l'intera rete, rimarrà per sempre o si estinguerà? In questo articolo indaghiamo alcune tecniche di modellizzazione epidemiologica per rispondere a queste domande. Nello specifico, siamo interessati a capire come si diffonde un virus in un network. Saranno necessari dei modelli analitici di propagazione virale, in grado di catturare l'impatto della topologia della rete su cui il virus si diffonde. Come verifica dei modelli teorici, verranno implementati 3 diversi tipi di network sui quali simulare un'epidemia e studiarne l'andamento.

Tutti i codici presentati in questo documento ed usati nelle simulazioni sono liberamente consultabili nella seguente repository di [GitHub](https://github.com/Videars/Epidemy-simulation)<sup>1</sup>.

---

<sup>1</sup><https://github.com/Videars/Epidemy-simulation>

## CAPITOLO 2

---

### Modelli stocastici e deterministici

---

Esistono due classi di modelli per analizzare fenomeni epidemici: i modelli stocastici e i modelli deterministici. Vediamone alcune differenze e cerchiamo di capire come affrontare il nostro studio.

Il vantaggio principale dei modelli deterministici risiede nella loro analisi più semplice: possono arrivare a contenere funzioni particolarmente complesse ma ancora possibili da analizzare, almeno quando hanno soluzioni numeriche adeguate. Al contrario, perché un modello epidemico stocastico sia matematicamente gestibile, deve essere abbastanza semplice e quindi non del tutto realistico.[1]

Tuttavia diverse ragioni suggeriscono che i modelli stocastici siano da preferire quando la loro analisi è possibile. In primo luogo, permettono una descrizione più dettagliata della struttura delle popolazioni studiate (attraverso l'utilizzo di grafi di contatto) e anche fenomeni che coinvolgono solo pochi individui o entità (per esempio, l'inizio e la fine di un'epidemia o i contatti in piccole unità locali), mentre i modelli deterministici, per lo più continui, descrivono fenomeni su più larga scala (analizzando porzioni più consistenti di una popolazione, con riferimento esplicito o implicito alla Legge dei Grandi Numeri) e spesso astraendo i schemi di contatto secondo la *Law of Mass Action*, cioè implicando un mescolamento omogeneo di individui. Vediamo più in dettaglio la sopra citata legge:

**Law of Mass Action.** *Assumiamo che la velocità con cui un'infezione si diffonde in una popolazione sia proporzionale al prodotto del numero di persone  $I$  che sono infettive con il numero di persone  $S$  che sono suscettibili all'infezione. Pertanto, se il tasso di nuove infezioni è  $C$ , la legge è scritta come:*

$$c = rIS$$

*dove  $r$  è una costante che descrive il grado di mescolamento tra infetti e suscettibili nella popolazione presa in esame.*

In effetti, una parte importante nella modellazione stocastica sta nel mostrare a quale modello deterministico converge il modello stocastico, quando la dimensione della popolazione diventa grande.

In secondo luogo, ci sono fenomeni che sono genuinamente stocastici e non soddisfano una legge dei grandi numeri. Alcune epidemie saranno responsabili della creazione di focolai di piccole dimensioni, infettando solo pochi individui, altre creeranno un focolaio importante, arrivando ad infettare una parte più o meno deterministica della comunità. Calcolare la probabilità dei due eventi è possibile solo in un contesto stocastico. Inoltre, quando si considera l'estinzione delle epidemie in stato endemico, questa può essere analizzata solo con modelli stocastici.

Un terzo importante vantaggio riguarda la stima. La conoscenza dell'incertezza nelle stime richiede un modello stocastico e una stima non è di grande utilità senza una nota incertezza associata. Per concludere, i modelli stocastici sono da preferire quando la loro analisi è possibile; altrimenti dovrebbero essere utilizzati modelli deterministici. I modelli deterministici possono anche fungere da modelli introduttivi nello studio di nuovi fenomeni. Non esiste alcun conflitto tra i due; anzi entrambi i tipi di modelli svolgono un ruolo importante per una migliore comprensione dei meccanismi di diffusione epidemici. Quello che faremo in questo paper è usare il modello deterministico classico come modello introduttivo per poi confrontarlo con alcuni modelli stocastici.

---

### Modello deterministico classico

---

#### 3.1 Classificazione della popolazione

Nei modelli deterministici si assumono delle ipotesi per semplificare la simulazione matematica della dinamica delle malattie infettive, sinteticamente tali ipotesi presumono che la popolazione sia divisa in compartimenti e che ogni individuo nello stesso compartimento abbia le stesse caratteristiche.

I modelli deterministici sono di solito costruiti con equazioni differenziali ordinarie che descrivono l'andamento, al variare del tempo, di quantità rilevanti per lo studio epidemico.

Il modello può essere utilizzato per prevedere le proprietà della diffusione di una malattia, ad esempio la prevalenza (numero totale di infetti) o l'incidenza di nuovi casi. Inoltre, il modello consente di comprendere in che modo diverse situazioni possono influenzare l'esito dell'epidemia, ad esempio quale sia la percentuale di vaccinazioni che in una determinata popolazione procura l'immunità di gregge oppure che variazione nel numero di riproduzione effettivo produce un contenimento dell'epidemia.

I modelli matematici più utilizzati necessitano di classificare la popolazione  $N$  in compartimenti. I più comunemente utilizzati sono:

- $S$ , suscettibili;
- $I$ , infetti;
- $E$ , esposti (usato quando la malattia richiede del tempo per rendere l'individuo infettivo);
- $R$ , recuperati, guariti, non infettabili perché immuni, dopo aver contratto la malattia.

Applicando il modello deterministico ad un'epidemia digitale possiamo delineare in modo analogo le classi precedenti nel seguente modo:

- $S$ , dispositivi nella rete che non sono stati infettati dal virus;
- $I$ , dispositivi nella rete che hanno il virus attivo in memoria e che quindi è pronto a diffondersi a tutti i dispositivi collegati;

- $E$ , dispositivi nella rete che hanno il virus inattivo in memoria e che quindi al momento non è in grado di diffondersi;
- $R$ , dispositivi nella rete che o possiedono software di protezione contro il virus o che lo hanno già avuto in memoria e li ha resi ormai inutilizzabili e non più elementi attivi nella diffusione dell'epidemia.

## 3.2 Parametri

Per lo sviluppo dei modelli sono poi in genere necessari:

- Il tasso di infezione o la velocità di trasmissione, normalmente indicato con  $\beta$ . Il suo reciproco  $1/\beta$  è il tempo medio tra i contatti. Nei modelli SIS o SIR il soggetto da suscettibile diventa infetto quando tra i due vi è un contatto adeguato alla trasmissione del virus.
- Il tasso di recupero, normalmente indicato con  $\mu$ . Il suo reciproco  $1/\mu$  è il tempo infettivo medio.
- Il tasso di latenza o di incubazione, normalmente indicato con  $\alpha$ . Il suo reciproco (indicato col simbolo  $\varepsilon$ ) è il tempo medio di latenza o incubazione, che intercorre da quando un soggetto diventa esposto a quando diventa infettivo.

## 3.3 SI

Il modello SI è un approccio particolarmente semplice per descrivere la diffusione delle malattie infettive dove  $N = I + S$  è costante. Il tasso di diffusione della malattia dipende statisticamente dal numero di individui infettivi e in secondo luogo dal numero di individui che possono ancora essere infettati. Secondo questo modello la malattia si diffonderà in tutta la popolazione. Le estensioni al modello SI sono il modello SIS, in cui gli individui possono guarire e riammalarsi, e il modello SIR, in cui gli individui possono diventare immuni alla malattia.

## 3.4 SIS

Il modello SIS, utilizzabile per malattie di breve durata che non danno immunità, distingue solo due gruppi di individui:  $S$  e  $I$ . Al momento  $t$  designato e con  $N$  costante, il numero di suscettibili  $S(t)$  e il numero di infettivi  $I(t)$  saranno in relazione:  $N = I(t) + S(t)$ . Il modello SIS può quindi essere utilizzato per le malattie che hanno le seguenti proprietà:

- Dopo che la malattia è guarita, ogni individuo ritorna immediatamente al gruppo sano e può essere nuovamente infettato.
- Gli infetti sono immediatamente contagiosi.
- Le persone sane si ammalano con un tasso di infezione lineare  $\beta$ .
- Le persone infette guariscono con un tasso di guarigione lineare  $\alpha$ .
- Ogni gruppo interagisce con la stessa probabilità. Ciò giustifica l'assunzione di relazioni lineari.



### 3.4.1 Formulazione deterministica e soluzione

La diffusione della malattia è generalmente formulata sotto forma di equazioni differenziali ordinarie:

$$\frac{dS}{dt} = -\beta IS + \mu I, \quad \frac{dI}{dt} = \beta IS - \mu I.$$

Per risolvere il sistema differenziale, sommiamo le due equazioni e otteniamo  $\frac{dS}{dt} + \frac{dI}{dt} = \frac{dN}{dt} = 0$  (relazione che seguiva già dalla costanza di  $N$ ). Sappiamo anche che  $S = N - I$ , allora  $\frac{dI}{dt} = (\beta N - \mu - \beta I)I$ . Chiamiamo  $\beta N - \mu = A$  ed isoliamo  $dI$  con  $dt$  ottenendo finalmente  $\frac{dI}{(A - \beta I)I} = dt$ . A questo punto basterà integrare e considerare la condizione iniziale  $I(0) = I_0$  ottenendo la soluzione esplicita:

$$I = \frac{(\beta N - \mu)I_0 e^{(\beta N - \mu)t}}{(\beta N - \mu) + \beta I_0 [e^{(\beta N - \mu)t} - 1]} \quad (3.1)$$

Studiamone l'andamento per avere un'idea su come si comporti l'epidemia.

Dalla soluzione del modello con  $t \rightarrow \infty$  si ricava che se  $\beta N - \mu < 0$  l'epidemia non ci sarà o si estinguerà, se invece  $\beta N - \mu > 0$  l'epidemia si evolverà tendendo ad un equilibrio tra numero di suscettibili e di infetti asintoticamente stabile, con un numero di suscettibili  $S_\infty = \frac{\mu}{\beta}$ , chiamato "equilibrio endemico".

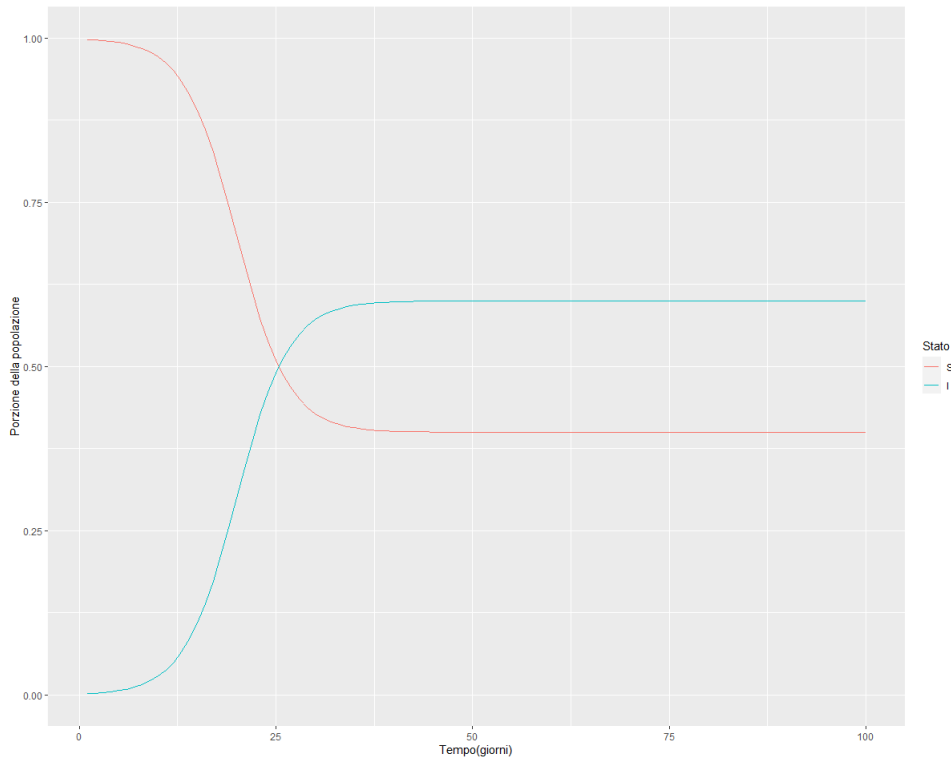


Figura 3.1: Soluzioni del modello SIS

Il grafico 3.1 mostra le due soluzioni del sistema di equazioni differenziali precedente nel caso in cui  $N = 100$ ,  $\beta = 1/2$  e  $\mu = 1/5$  (valori che torneranno nelle successive simulazioni). Si può notare che, poiché  $\beta N - \mu > 0$ , l'epidemia non si estingue ma raggiunge una situazione di stabilità con un numero di suscettibili fisso al livello  $\frac{\mu}{\beta}$  (nel

nostro caso corrispondente ai 2/5 della popolazione).

Questo vuol dire che all'inizio della diffusione (quando  $S \sim N$ ) ogni infetto infetterà circa  $\beta N/\mu$  suscettibili (mentre all'equilibrio endemico ogni infetto infetterà in totale 1 suscettibile).

Nel caso in cui invece definissimo:

$$\frac{dS}{dt} = -\beta IS/N + \mu I, \quad \frac{dI}{dt} = \beta IS/N - \mu I.$$

otterremmo la soluzione:

$$\frac{I}{N} = \frac{(\beta - \mu) \frac{I_0}{N} e^{(\beta - \mu)t}}{(\beta - \mu) + \frac{\beta I_0}{N} [e^{(\beta - \mu)t} - 1]} \quad (3.2)$$

la quale possiamo riscrivere come

$$\frac{I}{N} = \frac{(1 - \frac{\mu}{\beta}) \frac{I_0}{N} e^{(\beta - \mu)t}}{(1 - \frac{\mu}{\beta}) + \frac{I_0}{N} [e^{(\beta - \mu)t} - 1]} \quad (3.3)$$

Chiamando  $\frac{I_0}{N} = \gamma$  otteniamo l'equazione (6.1).

### 3.4.2 Relazione tra $\beta$ e $N$

Se il tasso di infezione della popolazione è definito come  $\beta SI$  (*density-dependent*) (con  $S$  e  $I$  i numeri di individui rispettivamente suscettibili ed infetti) vorrebbe dire che ogni infetto potrebbe riuscire a trasmettere il virus a  $\beta S$  individui per unità di tempo. Dunque, se  $\beta$  rimanesse invariato, ma venisse applicato ad un modello dove  $N$  è più grande, il numero di individui infettati da ogni infetto aumenterebbe.

Se invece si definisse il tasso d'infezione  $\beta IS/N$  (*frequency-dependent*), questo non succederebbe. La costante  $\beta$  può allora essere interpretata come numero di individui infettati da un infetto per unità di tempo, indipendentemente dalla grandezza della popolazione, se tutti gli individui contattati sono suscettibili. Questo numero viene poi modificato nel corso dell'epidemia dal fattore  $S/N$ , poichè solo questa proporzione degli individui contattati risulterà suscettibile.

Il punto chiave per le malattie umane è quindi determinare se il numero di contatti utili per trasmettere il virus tra due individui (uno infetto e l'altro suscettibile) cambia al variare della dimensione della popolazione. Per esempio, se si pensa al numero di contatti tra individui nelle città, un modello realistico è che questo numero sia più grande in una metropoli e più piccolo in una cittadina di campagna.

Ha quindi senso fare lo stesso ragionamento nelle epidemie digitali. I networks con un numero molto elevato di dispositivi (Internet per esempio) avranno sicuramente un numero di contatti tra gli stessi dispositivi notevolmente più elevato rispetto ad una rete locale.

In questa sezione si riassumeranno le principali definizioni e proprietà delle reti, rilevanti per studiare la diffusione delle epidemie.[3]

### 4.1 Definizioni generali

Le reti sono descritte matematicamente come grafi. Un grafo è un insieme di punti, chiamati vertici o nodi, uniti da un insieme di connessioni chiamate diversamente in base alla materia sulla quale stiamo usando la teoria dei grafi. Nella letteratura generalmente troveremo termini quali arco, lato, spigolo, bordo, collegamento o legame. Ogni bordo denota la presenza di una relazione o interazione tra i vertici che unisce.

I bordi possono rappresentare un'interazione bidirezionale tra vertici, o indicare una precisa direzionalità nell'interazione. Nel primo caso si parla di reti non orientate, nel secondo caso di reti orientate o digrafi. Da un punto di vista epidemiologico, la direzionalità di una rete è rilevante poiché impone restrizioni sui possibili percorsi di propagazione del contagio. Un modo compatto per elencare tutte le connessioni presenti in un grafo di dimensione  $N$  (cioè con  $N$  vertici) è la matrice  $N \times N$  di adiacenza  $A$ , con elementi  $a_{ij} = 1$  se i nodi  $i$  e  $j$  sono collegati da un arco, 0 altrimenti.  $A$  è simmetrica nei grafi non orientati e asimmetrica in grafi orientati. Un percorso  $P_{i_0, i_n}$  che collega i vertici  $i_0$  e  $i_n$  è una sequenza di archi diversi  $\{(i_j, i_{j+1})\}$ ,  $j = 0, \dots, n - 1$ . La *lunghezza* del percorso è il numero di archi attraversati.

Un grafo si dice *connesso* se, per ogni coppia di vertici, esiste un percorso che li collega. Un *componente*  $C$  di un grafo è definito come un sottografo connesso. Il *Giant component* è il componente o sottografo la cui dimensione scala come il numero di vertici nel grafico. Da un punto di vista epidemiologico, un virus che si diffonde nella componente gigante può infettare una frazione macroscopica del grafo, mentre se la malattia inizia al di fuori del componente gigante, il numero totale di vertici infetti sarà necessariamente limitato, rappresentando una frazione che decresce al crescere della dimensione della rete.

## 4.2 Metrica delle reti

### 4.2.1 Percorso più corto e diametro

La distanza  $l_{ij}$  tra i nodi  $i$  e  $j$  è definita come la lunghezza del percorso più breve (non necessariamente unico) che unisce  $i$  e  $j$ . Il *diametro* di una rete è il massimo valore di tutti i percorsi più brevi che collegano le coppie di nodi. Infine con  $\langle l \rangle$  si indicherà il valor medio di  $l_{ij}$  per tutte le coppie  $i$  e  $j$  di nodi.

### 4.2.2 Grado

Il grado  $k_i$  del vertice  $i$  in una rete non orientata è il numero di archi collegati ad  $i$ , cioè  $k_i = \sum_j a_{ij}$ . Nelle reti non orientate definiamo la distribuzione dei gradi  $P(k)$  come probabilità che un vertice generico abbia grado  $k$ , o, in reti finite, come frazione di vertici nel grafico con grado esattamente uguale a  $k$ . È utile considerare i momenti di distribuzione dei gradi  $\langle k^n \rangle = \sum_k k^n P(k)$ . Il momento primo ( $n = 1$ ), ovvero il grado medio  $\langle k \rangle = 2L/N$  con  $L$  il numero totale di archi, fornisce informazioni sulla densità della rete. Una rete si dice *sparsa* se  $L$  cresce al massimo linearmente con  $N$ ; altrimenti si chiama *denso*.

### 4.2.3 Correlazione tra gradi

Le correlazioni di grado tra due vertici possono essere misurate mediante la probabilità condizionata  $P(k'|k)$  che un arco di un vertice di grado  $k$  sia connesso ad un vertice di grado  $k'$ . Una rete si dice *non correlata* (*uncorrelated*) se questa probabilità è indipendente dal vertice di grado  $k$ . In questo caso,  $P(k'|k)$  può essere stimata come rapporto tra il numero di archi che puntano a vertici di grado  $k'$ ,  $k'P(k')N/2$ , e il numero totale di archi,  $\langle k \rangle N/2$ , ottenendo  $P^{un}(k'|k) = \frac{k'P(k')}{\langle k \rangle}$ .

### 4.2.4 Centralità

Il concetto di centralità ha il fine di valutare l'importanza di un nodo all'interno di una rete. Esistono diverse definizioni di centralità in base ai valori presi in esame. La più semplice tra tutte è la centralità di grado. Maggiore è il grado, più il nodo sarà considerato influente/centrale nella rete (dunque rilevante per l'evoluzione epidemica). Le definizioni alternative si basano sui cammini minimi tra i vertici. La *closeness centrality*  $C_i$  (centralità di vicinanza) è definita come l'inverso della media delle lunghezze dei cammini minimi dal vertice  $i$  a tutti gli altri vertici nella rete. Con questa definizione, consideriamo un vertice centrale se è situato, in media, a breve distanza a tutti gli altri vertici della rete.

## 4.3 Classi di reti e modelli

La recente abbondanza di dati e misurazioni di reti del mondo reale ha evidenziato l'esistenza di diverse classi di reti, caratterizzate da una grande variabilità nelle metriche e nelle proprietà statistiche. Di seguito sono elencate alcune delle principali classi di rete utilizzate per esplorare le proprietà dei processi epidemici.

### 4.3.1 Networks omogenei

Il primo modello teorico di reti è il modello di grafo casuale classico. Nella sua formulazione più semplice, il grafo  $G_p(N)$  si costruisce da un insieme di  $N$  nodi in cui ciascuno degli  $N(N-1)/2$  possibili collegamenti è presente con probabilità  $p$ . La distribuzione dei gradi è data da una forma binomiale che per  $p = \langle k \rangle / N$  e  $N$  molto grande può essere approssimata da una distribuzione di Poisson  $P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$ . In più ha un picco intorno al valor medio, denotando così una omogeneità statistica dei nodi.

### 4.3.2 Small-World networks

Il modello *Small-World* considera un reticolo ordinato, come un anello di  $N$  vertici, in cui ogni nodo è connesso simmetricamente ai suoi  $2m$  nodi più vicini. Partendo da questa prima configurazione, una frazione  $p$  di archi viene ricablata, visitando tutti gli  $m$  bordi in senso orario di ciascun vertice e riconnettendoli, con probabilità  $p$ , a un nodo casualmente scelto. L'effetto complessivo del processo di ricablaggio è quello di aggiungere scorciatoie a lungo raggio che, anche per un piccolo valore di  $p \sim N^{-1}$ , riduce notevolmente la lunghezza media del percorso più breve. Questo modello ha una distribuzione dei gradi e misure di centralità che decadono con velocità esponenziale allontanandosi dal valore medio.

### 4.3.3 Heavy-tailed networks

L'evidenza empirica ha mostrato che le reti del mondo reale sono strutturate in una gerarchia di nodi: alcuni aventi connettività molto grande (gli *hub*) altri invece, la grande maggioranza dei nodi, aventi gradi più piccoli. Diversamente da quanto visto nei reticoli regolari ed omogenei, in cui un generico nodo avrà un grado prossimo al grado medio, le reti eterogenee mostrano una *Heavy-tailed degree distribution* spesso approssimata da una *Power law* della forma  $P(k) \sim k^{-\gamma}$ . Questo implica che c'è una probabilità non trascurabile di trovare vertici con gradi molto elevati.

Un network di questo tipo si può implementare partendo dal presupposto che le reti osservabili nel mondo reale non hanno un numero di lati costanti ma variabili nel tempo con l'inserimento di nuovi nodi. Prendiamo per esempio in esame il modello di Barabasi-Albert (che useremo poi nella modellizzazione dell'epidemia). Si implementa nel seguente modo:

- (i) La rete inizia con un piccolo nucleo di  $m_0$  vertici connessi; ad ogni passo viene aggiunto un nuovo nodo, con  $m$  ( $m \leq m_0$ ) archi che sono collegati ai vecchi vertici nella rete.
  - (ii) I nuovi archi sono connessi all' $i$ -esimo nodo con probabilità pari a  $F(k_i) = k_i / \sum_j k_j$ .
- In un tempo sufficientemente lungo, il network generato ha una *degree distribution*  $P(k) \sim k^{-3}$ .

### 4.3.4 Reti statiche e dinamiche

Finora abbiamo assunto che l'insieme dei nodi e dei collegamenti non cambi nel corso del tempo (i networks generati sotto questo assunto si diranno statici). Tuttavia, molte reti reali sono tutt'altro che statiche. I loro collegamenti vengono creati, distrutti e ricablati continuamente e seguendo regole diverse. In alcune di queste reti dinamiche, come Internet, la scala temporale dell'evoluzione della rete è piuttosto lenta. Una rete statica quindi fornisce una buona approssimazione di reti dinamiche quando le proprietà del fenomeno

che si vanno ad analizzare variano molto più velocemente dell'evoluzione temporale della rete. Il limite opposto definisce le cosiddette *annealed networks*, che descrivono il caso in cui l'evoluzione della rete è molto più veloce dei processi dinamici.

---

## Processo Markoviano

---

Si definisce processo stocastico markoviano (o di Markov), un processo aleatorio in cui la probabilità di transizione che determina il passaggio a uno stato di sistema dipende solo dallo stato del sistema immediatamente precedente (proprietà di Markov) e non da come si è giunti a questo stato. Viceversa si dice processo non markoviano un processo aleatorio per cui non vale la proprietà di Markov. Un processo di Markov può essere descritto per mezzo dell'enunciazione della proprietà di Markov, o condizione di "assenza di memoria", che può essere scritta come:

$$P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0) = P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n) \quad (5.1)$$

### 5.1 Catene di Markov

Una catena di Markov è un processo di Markov con spazio degli stati discreto, quindi è un processo stocastico che assume valori in uno spazio discreto e che gode della proprietà di Markov. L'insieme  $S$  di spazio degli stati può essere finito o infinito numerabile. Nel primo caso si parla di catena di Markov a stati finiti. Una catena di Markov può essere tempo-continua o tempo-discreta, in base all'insieme  $T$  di appartenenza della variabile tempo (continuo o discreto).

Formalmente una catena di Markov è un processo stocastico Markoviano caratterizzato da un parametro  $t_i \in T$ , da un insieme  $S$  di stati e da una funzione probabilità di transizione  $P : S \times S \times T \mapsto [0, 1]$ .

Essendo un processo Markoviano,  $P$  gode della proprietà di Markov (5.1).

Nel caso di catena di Markov a tempo discreto, cioè con l'insieme  $T$  discreto, la notazione si semplifica e otteniamo:

$$P(X_{n+1} = x_{n+1} | X_n, X_{n-1}, \dots, X_0) = P(X_{n+1} = x_{n+1} | X_n).$$

Una catena di Markov omogenea è un processo markoviano in cui la probabilità di transizione al tempo  $t_i$  non dipende dal tempo stesso, ma soltanto dallo stato del sistema al tempo immediatamente precedente  $X(t_{i-1})$ . In altre parole, la probabilità di transizione è indipendente dall'origine dell'asse dei tempi e quindi dipende soltanto dalla distanza tra i

due istanti temporali.

Per le catene omogenee vale la condizione:

$$P(X_{n+1} = x \mid X_n = y) = P(X_n = x \mid X_{n-1} = y)$$

## 5.2 Matrice di transizione

Una catena di Markov omogenea a stati finiti in cui l'insieme  $S$  degli stati del sistema è finito e ha  $N$  elementi può essere rappresentata mediante una matrice di transizione  $A \in \mathbb{R}^{N \times N}$  e un vettore di probabilità iniziale  $\tilde{\pi}_0 \in \mathbb{R}^N$ .

Gli elementi di  $A$  rappresentano le probabilità di transizione tra gli stati della catena: una catena che si trova nello stato  $i$  ha probabilità  $a_{ij}$  di passare allo stato  $j$  nel passo immediatamente successivo. In particolare gli elementi  $a_{ii}$  sulla diagonale principale di  $A$  indicano le probabilità di rimanere nello stesso stato  $i$ . Il vettore  $\tilde{\pi}_0$  definisce le probabilità che inizialmente la catena di Markov si trovi in ciascuno degli  $N$  stati. Una catena di Markov omogenea è univocamente definita dalla coppia  $(A, \tilde{\pi}_0)$ .

Se al tempo  $t_0$  ha la distribuzione di probabilità  $\tilde{\pi}_0$  allora le probabilità  $\tilde{\pi}_n$  che, ad un tempo  $t_n$  (dopo  $n$  transizioni), il sistema si trovi in ciascuno degli  $N$  stati, sono date dal vettore così definito:

$$(\tilde{\pi}_n)^T = (\tilde{\pi}_{n-1})^T \cdot A$$

dove  $(\tilde{\pi}_n)^T$  indica la trasposta del vettore  $\tilde{\pi}_n$ . Iterando il procedimento arriviamo a:

$$(\tilde{\pi}_n)^T = (\tilde{\pi}_0)^T \cdot A^n$$

Dalla definizione assiomatica della probabilità discendono le seguenti proprietà per la matrice  $A$ :

- $a_{ij} \geq 0, \quad \forall i, j \in \{1 \dots N\}$
- $\sum_{j \in S} a_{ij} = 1, \quad \forall i \in \{1 \dots N\}$

La seconda proprietà equivale a richiedere che la somma degli elementi su ciascuna riga sia uguale a 1.

Nel caso di una catena di Markov omogenea a stati discreti si può adottare la notazione sintetica:

$$P_{ij}^{(n)} \equiv P(X_{m+n} = s_j \mid X_m = s_i) = P(X_n = s_j \mid X_0 = s_i)$$

dove  $(n)$  non è un esponente bensì un indice.

Si ha quindi  $(\tilde{\pi}_n)_j = \sum_{i \in S} (\tilde{\pi}_0)_i (P^{(n)})_{ij}$ .

Si hanno le seguenti proprietà:

- $P_{ij} \geq 0 \quad \forall i, j \in S$
- $\sum_{j \in S} P_{ij} = 1$

## 5.3 Catene di Markov aperiodiche

Il periodo di uno stato  $s_i \in S$  di una catena di Markov a stati discreti, con  $S$  finito o infinito numerabile, è definito come il minimo numero di passi temporali affinché vi sia una probabilità diversa da zero di tornare sullo stesso stato, partendo dallo stato  $s_i$  al



tempo  $t_m$ .

Formalmente il periodo  $d(s_i)$  è definito come segue:

$$d(s_i, t_m) = MCD\{n \geq 1 : P(X_{m+n} = s_i | X_m = s_i) > 0\}$$

dove  $MCD$  indica il massimo comune divisore.

Nel caso di una catena di Markov omogenea a stati finiti con numero di stati  $N$ , rappresentabile quindi con una matrice  $A \in \mathbb{R}^N$ , si può riformulare la definizione così:

$$d(s_i) = MCD\{n \geq 1 : (A^n)_{ii} > 0\}.$$

Lo stato  $s_i$  è detto aperiodico se il suo periodo è uguale a 1.

Una catena di Markov è detta aperiodica se tutti i suoi stati sono aperiodici.

## 5.4 Catene di Markov irriducibili

Una catena di Markov a stati discreti è detta irriducibile se partendo da ogni stato  $s_i$  c'è una probabilità maggiore di zero di raggiungere ogni altro stato  $s_j$ . Formalmente una catena di Markov è irriducibile se:

$$\forall s_i, s_j \in S, \forall m \in \mathbb{N}, \exists n \in \mathbb{N} : P(X_{m+n} = s_j | X_m = s_i) > 0.$$

## 5.5 Stati ricorrenti positivi

Sia

$$T_i = \begin{cases} \min\{n \geq 1 \text{ t.c. } X_n = s_i\} & \text{se } X_n = s_i \text{ per qualche } n \geq 1 \\ +\infty & \text{altrimenti} \end{cases}$$

lo stato  $s_i \in S$  si dice ricorrente positivo se  $E(T_i | X_0 = s_i) = \mu_i < +\infty$ .

Se una catena è irriducibile e un suo stato è ricorrente positivo allora tutti i suoi stati sono ricorrenti positivi, in tale caso la catena si dice ricorrente positiva.

## 5.6 Distribuzioni stazionarie

Data una catena di Markov omogenea a stati discreti, una sua distribuzione stazionaria di probabilità, detta anche distribuzione di equilibrio,  $\pi = \{\pi_1 \dots \pi_n \dots\}$  è una distribuzione discreta di probabilità che soddisfa le seguenti:

- $\forall i \in S, \pi_i \geq 0$
- $\sum_{i \in S} \pi_i = 1$
- $\forall j \in S, \sum_{i \in S} \pi_i P_{ij} = \pi_j$

Informalmente una distribuzione stazionaria è una distribuzione di probabilità che si mantiene costante all'evolversi nel tempo della catena di Markov.

---

## Modello stocastico SIS semplice

---

Il modello stocastico SIS per le infezioni endemiche è definibile nel seguente modo.[1] Supponiamo di avere una popolazione omogenea e chiusa (priva di altri contatti con l'esterno) di  $n$  individui. Facciamo sì che  $m$  di questi individui diventino infetti al tempo  $t = 0$ . Ogni individuo infetto rimane tale per un periodo temporale che va come una distribuzione esponenziale di parametro  $\mu$  e, durante questo intervallo, l'individuo entra in contatto con gli altri individui della popolazione ad un tasso costante di  $\beta/n$ . Se un individuo contattato è suscettibile, diventa immediatamente infetto. Un individuo infetto diventa di nuovo suscettibile appena passato il periodo di infezione. Si presume che tutti i periodi infettivi e il numero di contatti (descritto da processi di Poisson) siano reciprocamente indipendenti. Poiché gli individui sono o suscettibili o infetti, al fine di studiare l'andamento epidemico, è sufficiente tenere traccia solo del numero di infetti. Denotiamo con  $Y(t)$  il numero di individui infetti al tempo  $t$ . Allora  $Y = \{Y(t); t \geq 0\}$ , può essere descritto come un semplice processo di nascita e morte in tempo continuo nello spazio degli stati  $S_n = \{0, \dots, n\}$ , con i seguenti *rate* di transizione:

$$\begin{array}{lll} \text{da} & a & \text{ad un tasso} \\ i & i + 1 & \beta i(n - i)/n \\ i & i - 1 & \mu i \end{array}$$

e con  $Y(0) = m$  e  $Y(t) = 0$  stato assorbente. Si noti che, per ridimensionare il tasso di infezione, invece di utilizzare solo il numero di suscettibili iniziali, stiamo utilizzando la dimensione totale della popolazione. Il motivo è che gli infetti iniziali possono diventare infetti più volte nel corso dell'epidemia e stanno così prendendo un ruolo più attivo nell'andamento epidemico.

### 6.1 Legge dei grandi numeri

Si consideri una sequenza  $Y_n$  di modelli epidemici, tutti con gli stessi parametri  $\beta$  e  $\mu$  e con  $n$  che indica la popolazione totale. Se il rapporto,  $m_n/n$ , che descrive la proporzione di infetti iniziale, tende al valore non triviale  $\gamma$ , con  $0 < \gamma < 1$  per  $n$  che tende all'infinito, per i processi  $\bar{Y}_n = Y_n/n$  vale la legge dei grandi numeri[1].

Concludiamo dunque che  $|\bar{Y}_n(t) - y(t)|$  tende a zero in probabilità uniformemente sugli intervalli temporali compatti per  $n \rightarrow \infty$ , dove  $y(t)$  è la soluzione dell'equazione differenziale

$$\frac{dy}{dt} = \beta y(1 - y) - \mu y$$

e  $y(0) = \gamma$ . la soluzione esplicita è data da

$$\begin{cases} y(t) = \frac{(1 - \frac{\mu}{\beta})\gamma e^{(\beta - \mu)t}}{1 - \frac{\mu}{\beta} + \gamma(e^{(\beta - \mu)t} - 1)} & \beta \neq \mu \\ y(t) = \frac{\gamma}{1 + \beta\gamma t} & \beta = \mu \end{cases} \quad (6.1)$$

Il numero di riproduzione di base (comunemente noto come  $R_0$ ) è dato da  $\beta/\mu$ . Notiamo che  $y(t) \rightarrow 0$  per  $t \rightarrow \infty$  se  $\beta/\mu \leq 1$ . Altrimenti se  $\beta/\mu > 1$  allora  $y(t) \rightarrow \hat{y} = 1 - \mu/\beta$  per  $t \rightarrow \infty$ . In questo caso, il valore  $n\hat{y}$  è chiamato livello endemico del processo. Questi risultati indicano che il modello stocastico SIS si comporterà in modo molto diverso a seconda che il numero di riproduzione di base è inferiore o superiore a 1.

Inoltre il suo comportamento, almeno su intervalli limitati di tempo, si può approssimare con il modello deterministico SIS.

## 6.2 Tempo all'estinzione

Procediamo con lo studiare il tempo  $T_n$  per l'estinzione del processo  $Y_n$

$$T_n = \inf\{t \geq 0 : Y_n(t) = 0\}$$

Poichè  $Y_n$  è irriducibile sull'insieme  $S_n \setminus \{0\}$  e ha  $\{0\}$  come stato di assorbimento, il tempo di estinzione è finito. Sarebbe interessante capire cosa succede a  $T_n$  per una popolazione che cresce all'infinito. Il teorema enunciato sotto mostra che il tempo all'estinzione per il modello stocastico è una quantità che riflette il comportamento soglia del processo deterministico. Nel caso in cui  $\beta \leq \mu$ , il tempo per l'estinzione è sempre relativamente breve, mentre nel caso  $\beta > \mu$  potrebbe essere molto grande.

Più nel dettaglio il teorema del tempo all'estinzione del modello SIS stocastico ha le seguenti 4 proprietà asintotiche (il teorema è enunciato formalmente nell'appendice A):

1. Se il numero di riproduzione di base  $R_0 = \beta/\mu$  è superiore a 1 e il numero di infettivi iniziali è grande, il tempo di estinzione  $T_n$  cresce esponenzialmente con la dimensione della popolazione.
2. Se  $R_0 > 1$  e iniziamo con un piccolo numero di infetti, allora si presenta una sorta di fenomeno soglia. Da una parte dello spazio campionario,  $T_n$  rimane piccolo, dall'altra  $T_n$  cresce esponenzialmente.
3. Se  $R_0 \leq 1$  ma il numero di infetti iniziali è grande, allora  $T_n$  si comporta come  $\log n$ .
4. Infine, se  $R_0 \leq 1$  e all'inizio c'è un piccolo numero di individui infetti, il tempo per l'estinzione è sempre piccolo.

In conclusione, il modello stocastico si comporterà come quello deterministico ma con una distinzione rilevante: il numero di infetti  $I$  presenterà delle fluttuazioni più o meno evidenti intorno alla soluzione del modello deterministico seguendo un comportamento chiamato *quasi-stationary distribution* per poi arrivare certamente allo stato di assorbimento  $I = 0$ .

---

## Mean Field Theory per Epidemie su Grafi

---

Un'importante generalizzazione del semplice modello SIS è il passaggio da contatti omogenei tra tutti gli individui a individui connessi in modo strutturato.

Consideriamo quindi una rete, definita dalla sua matrice di adiacenza  $A$ , contenente tutti i possibili contatti tra i dispositivi del network preso in esame. Assumiamo che lo stato del nodo  $i$  (ovvero l' $i$ -esimo dispositivo nella rete) sia determinato dalla variabile aleatoria  $X_i(t) \in 0, 1, \dots, q-1$ , dove  $X_i(t) = a$  vuol dire che il nodo  $i$  appartiene allo stato  $a$  nel tempo  $t$ . Assumiamo che tutte le transizioni tra stati siano descritte da processi di Poisson indipendenti. Sotto queste ipotesi, l'evoluzione dell'epidemia può essere visualizzata come una catena di Markov. In una rete con  $N$  nodi, il numero totale di stati è  $q^N$ , ovvero tutte le possibili combinazioni in cui gli  $N$  nodi possono assumere un valore da 0 a  $q-1$ . Si possono esplicitamente calcolare gli elementi del generatore infinitesimale  $q^N \times q^N$  della catena di Markov che chiameremo  $Q$ . Una volta noti  $Q$  e le probabilità di infezione iniziali, si potranno determinare le probabilità  $Pr[X_1(t) = x_1, \dots, X_N(t) = x_N]$  al tempo  $t$  per ogni  $x_j = 0, 1, \dots, q-1$  usando ordinarie operazioni su matrici. Nonostante questo approccio sia esatto, viene usato raramente a causa delle seguenti motivazioni. Determinare  $Q$  richiede la risoluzione di un sistema lineare con  $q^N \times q^N$  equazioni (e ciò ne limita l'applicazione a reti con pochi nodi e/o stati). Inoltre la struttura stessa di  $Q$  è piuttosto complessa e questo non ci permette di trarre informazioni utili senza uno studio ulteriore. Andremo perciò ad illustrare le basi di un altro approccio teorico nello studio di epidemie che si basa sulla teoria *mean-field*. [4]

### 7.1 IBMF

L'*Individual-Based Mean Field theory* (IBMF) rappresenta una drastica semplificazione del modello sopra descritto. L'idea di base sta nel rappresentare l'evoluzione della probabilità  $\rho_i^a(t)$  che il nodo  $i$  appartenga allo stato  $a$ , per ogni nodo  $i$ , assumendo che lo stato di ogni nodo sia statisticamente indipendente dallo stato dei nodi immediatamente vicini. Le *mean-field equations* possono essere ottenute, sotto la precedente assunzione, applicando un'estensione della *law of mass action* (già introdotta nel capitolo 2), per esempio assumendo che la probabilità che il nodo  $i$  nello stato  $a$  e un suo adiacente  $j$  nello stato  $b$  sia  $\rho_i^a(t) \cdot \rho_j^b(t)$ . Sistematicamente, queste equazioni possono essere ottenute dalle

equazioni che regolano i  $q^N$  stati della catena di Markov, assumendo che i valori medi di coppie di variabili si fattorizzino nel seguente modo:  $E[X_i X_j] = E[X_i]E[X_j]$ . Le soluzioni di un modello che usa l'IBMF dipendono in generale dallo spettro della matrice di adiacenza del grafo associato alla rete, in particolare dall'autovalore di modulo massimo che chiameremo  $\Lambda$ .

## 7.2 Approccio IBMF sul modello SIS

Prendiamo un insieme di variabili aleatorie Bernoulli  $X_i(t) \in \{0, 1\}$ :  $X_i(t) = 0$  per un nodo suscettibile e  $X_i(t) = 1$  per un nodo infetto. Allora  $E[X_i] = \Pr[X_i = 1] \equiv \rho_i^I(t)$ . Questo ci permette di scrivere la seguente equazione che regola i modelli SIS:

$$\frac{dE[X_i(t)]}{dt} = E \left[ -\mu X_i(t) + (1 - X_i(t))\beta \sum_{j=1}^N a_{ij} X_j(t) \right] \quad (7.1)$$

La (7.1) descrive la variazione nel tempo della probabilità di infezione del nodo  $i$  e corrisponde al valor medio di due variabili aleatorie contrapposte: (a) se il nodo  $i$  è infetto ( $X_i = 1$ ), allora  $\frac{dE[X_i(t)]}{dt}$  diminuisce ad un ritmo pari al tasso di guarigione  $\mu$  e (b) se il nodo è sano ( $X_i = 0$ ), può essere infettato (con un tasso di infezione pari a  $\beta$ ) dai nodi infetti vicini. Il numero totale di infetti vicini al nodo  $i$  è  $\sum_{j=1}^N a_{ij} X_j$ . Ora per network statici la (7.1) diventa:

$$\frac{d\rho_i^I(t)}{dt} = -\rho_i^I(t) + \lambda \sum_{j=1}^N a_{ij} \rho_j^I(t) - \lambda \sum_{j=1}^N a_{ij} E[X_i(t)X_j(t)] \quad (7.2)$$

dove  $t$  è stato riscalato di  $1/\mu$  e  $\lambda = \beta/\mu$ . Il fatto che ci sia dipendenza tra nodi, ci porta necessariamente ad applicare l'assunto del modello IBMF considerando

$$E[X_i(t)X_j(t)] = E[X_i(t)]E[X_j(t)] = \rho_i^I(t)\rho_j^I(t) \quad (7.3)$$

Sotto quest'approssimazione le equazioni (7.2) diventano:

$$\frac{d\rho_i^I(t)}{dt} = -\rho_i^I(t) + \lambda[1 - \rho_i^I(t)] \sum_{j=1}^N a_{ij} \rho_j^I(t) \quad (7.4)$$

Studiamo adesso alcuni modi per trovare le soluzioni stazionarie del sistema (7.4), cioè quando tutte le derivate sono poste uguali a 0. Iniziamo con l'illustrare il seguente risultato:[6]

Consideriamo il grafo  $\mathcal{G}(\mathcal{N}, \mathcal{E})$  e assumiamo che per ogni dato tempo  $t \geq 0$ , ogni nodo  $i \in \mathcal{N}$  del grafo è o in uno stato infetto ( $I$ ) o suscettibile ( $S$ ). Un nodo infetto può infettare tutti i suoi vicini suscettibili ad un *rate* fisso che chiameremo  $\beta > 0$ . Può anche tornare nello stato suscettibile ad un *rate* fissato chiamato  $\mu > 0$ . Scriveremo  $x(t) = [x_i(t)] \in \mathbb{R}^N$ , dove  $x_i(t)$  rappresenta la probabilità che il nodo  $i \in \mathcal{N}$  sia infetto ad un tempo  $t \geq 0$ . Allora la dinamica del modello SIS può essere descritta attraverso il sistema di ODEs seguente:

$$\frac{dx_i(t)}{dt} = f_i(x(t)) = \beta(1 - x_i(t)) \sum_{j \in \mathcal{N}} a_{ij} x_j(t) - \mu x_i(t) \quad (7.5)$$

per ogni  $i \in N$  e  $t \geq 0$ . Possiamo riscrivere il sistema in forma matriciale:

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}) = \beta \cdot \text{diag}(\mathbf{1} - \mathbf{x})\mathbf{A}\mathbf{x} - \mu\mathbf{x} \quad (7.6)$$

dove abbiamo sottointeso la notazione  $(t)$  e con  $\mathbf{A}$  indichiamo la matrice di connettività associata al grafo  $\mathcal{G}$ . Il sistema (7.6) è positivamente invariante nell'insieme  $[0, 1]^N$  e ha  $\mathbf{0}$  come uno dei suoi punti fissi.

**Teorema 1.** Poniamo  $\lambda = \beta/\mu$  e sia  $\Lambda(\mathbf{A})$  l'autovalore di massimo modulo di  $\mathbf{A}$ . Allora

1. o  $\lambda \leq 1/\Lambda(\mathbf{A})$  e  $\mathbf{x}^* = \mathbf{0}$  è un punto fisso asintoticamente stabile di (7.6);
2. oppure  $\lambda > 1/\Lambda(\mathbf{A})$  ed esiste un unico punto fisso strettamente positivo  $\mathbf{x}^* \in (0, 1)^N$  che sia globalmente asintoticamente stabile in  $[0, 1]^N \setminus \{\mathbf{0}\}$ .

Per il nostro caso basta porre  $x_i(t) = \rho_i^I(t)$ .

Avendo a disposizione il teorema (1) che ci garantisce l'esistenza di uno zero asintoticamente stabile, illustriamo il metodo di Newton per funzioni vettoriali [7] e applichiamo al sistema (7.4) nel caso stazionario (ponendo  $\frac{d\mathbf{x}}{dt} = 0$ ) e in dimensione  $N = 2$  per determinarne gli zeri.

Il problema che vogliamo risolvere è quindi:

$$\begin{cases} 0 = -x + \lambda(1 - x)y \\ 0 = -y + \lambda(1 - y)x \end{cases} \quad (7.7)$$

con  $x = \rho_1^I$  e  $y = \rho_2^I$ . (7.7) è stato ottenuto determinando la matrice di connettività per il grafo elementare a due nodi reciprocamente connessi (quindi la matrice  $2 \times 2$  con tutti 1 sull'antidiagonale) e sostituendone i coefficienti in (7.4).

Notiamo che il sistema (7.7) può essere risolto analiticamente. Infatti definendo la funzione  $g(t) = t/(\lambda(1 - t))$ , possiamo esprimere  $y = g(x)$  e  $x = g(y)$ . A questo punto, calcolando i punti fissi di  $g(g(t))$  in  $[0, 1]$  (che hanno la forma  $(0, 0)$  e  $(1 - 1/\lambda, 1 - 1/\lambda)$ ), avremo determinato gli zeri del sistema in questione.

Creiamo comunque un piccolo script in *MatLab* per eseguire le approssimazioni del metodo di Newton, in modo tale da poterlo applicare al sistema (7.4) in situazioni in cui la risoluzione analitica non è implementabile.

```

1 %Parameters
2 Beta=0.5;
3 Mu=0.2;
4 Lam=Beta/Mu;
5
6 % Functions
7 f = @(xy) [-xy(1) + Lam*(1-xy(1))*xy(2);
8           -xy(2) + Lam*(1-xy(2))*xy(1)];
9 % Jacobian of f
10 J = @(xy) [1 - Lam*xy(2), Lam*(1-xy(1));
11           Lam*(1-xy(2)), 1 - Lam*xy(1)];
12 % Jacobian_inverse of f

```

```

13 Jinv = @(xy) (1/((1+Lam*xy(2))*(1+Lam*xy(1))-(Lam)^2*
14 (1-xy(1))*(1-xy(2))))*[(-1-Lam*xy(1)), (Lam*(xy(1)-1));
15 (Lam*(xy(2)-1)), (-1-Lam*xy(2))];
16 % Initial guesses
17 xy = [0.8; 0.8];
18 xy = newton2d(f,J,xy);
19
20 x = xy(1); y = xy(2);
21
22 disp('parameter_values:')
23 disp(['x=', num2str(x), 'y=', num2str(y)])
24
25 disp('function_values:')
26 disp(f(xy))

```

Nel *main* definiamo i parametri, le due funzioni, lo jacobiano, il suo inverso e il vettore iniziale dal quale far partire le iterazioni. Richiamiamo poi la funzione *newton2d* che andrà effettivamente ad eseguire le approssimazioni.

```

1 function xy = newton2d(f,J,xy)
2
3     tol = 10^-6;
4     err = 1;
5     maxits = 100;
6     its = 0;
7     while (err>tol) && (its<maxits)
8         xyold = xy;
9         xy = xyold - J(xyold)*f(xyold);
10        err = norm(xy-xyold);
11        its = its+1;
12    end
13
14    disp('iterations:')
15    disp(its)
16
17    disp('error:')
18    disp(err)
19 end

```

L'output finale sarà:

```

iterations: 5
error: 1.9756e-10
zeroes values: x = 0.6 y = 0.6
function values: (0,0)

```

Notiamo la presenza di un punto di equilibrio asintoticamente stabile in (0.6, 0.6) e di un punto di equilibrio instabile in (0, 0). Ci troviamo infatti nel caso 2 del teorema (1) poichè il nostro  $\lambda = 5/2 > 1/\Lambda(A) = 1$ . Possiamo anche verificarlo graficamente con lo *slope field* del sistema 7.7 nel caso non stazionario.

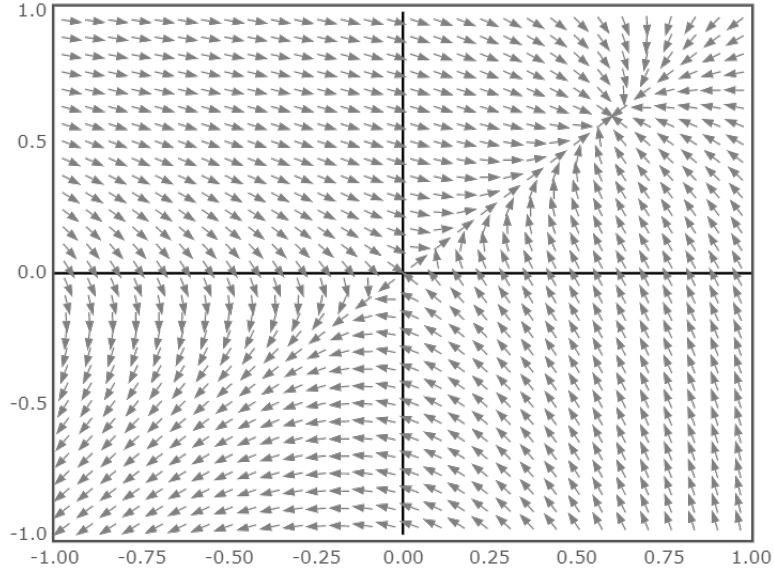


Figura 7.1: Slope field di 7.7

Ovviamente a noi basterà considerare il comportamento nel II quadrante della figura (7.1) poichè le quantità considerate sono probabilità e dunque appartenenti a  $[0, 1]^N$ .

Per ottenere informazioni sulla soglia dell'epidemia basterà linearizzare (7.4) ottenendo lo Jacobiano con i rispettivi elementi  $J_{ij} = -\delta_{ij} + \lambda a_{ij}$ . Lo stato endemico si verifica quando l'autovalore di modulo massimo ( $\Lambda$ ) è positivo e quindi

$$\lambda \geq \lambda_c^{IBMF}, \quad \lambda_c^{IBMF} = \frac{1}{\Lambda} \quad (7.8)$$

In reti i cui gradi hanno una distribuzione power law  $P(k) \sim k^{-\gamma}$ , l'equazione (7.8) può essere combinata con  $\Lambda \sim \max\{\sqrt{k_{max}}, \langle k^2 \rangle / \langle k \rangle\}$ , dove  $k_{max}$  è il massimo grado della rete[4]. Produciamo così un'espressione dell'andamento della soglia epidemica:

$$\lambda_c^{IBMF} \simeq \begin{cases} 1/\sqrt{k_{max}} & \gamma > 5/2 \\ \langle k^2 \rangle / \langle k \rangle & 2 < \gamma < 5/2 \end{cases} \quad (7.9)$$



---

### Simulazione di epidemie SIS deterministiche e stocastiche

---

In questo capitolo creeremo tre diversi tipi di Network in cui poter diffondere un virus e studiare l'andamento dell'epidemia generata. La simulazione è stata implementata attraverso il linguaggio di programmazione R e con l'ausilio di librerie necessarie per la risoluzione di equazioni differenziali, generazione di networks con le proprietà richieste e la rappresentazione grafica dei network generati.

Qui sotto sono elencate nel dettaglio le librerie usate:

```
1 library(igraph)
2 library(ggplot2)
3 library(deSolve)
4 library(reshape2)
5 library(R.matlab)
6 library(Matrix)
```

1. **igraph** può essere utilizzato per generare grafi, calcolare misure di centralità e proprietà basate sulla lunghezza del percorso, nonché identificare componenti del grafo. Può leggere e scrivere file Pajek e GraphML, nonché semplici *edge lists*.
2. **ggplot2** è una libreria per creare grafici in R. Basta fornire alla funzione *ggplot* i dati numerici e le variabili estetiche che ci interessano.
3. **deSolve** è un pacchetto che permette di risolvere problemi ai valori iniziali di un sistema di equazioni differenziali ordinarie del primo ordine, di equazioni differenziali alle derivate parziali, di equazioni differenziali algebriche e di equazioni differenziali di *delay*.
4. **reshape2** è un pacchetto di R che semplifica la trasformazione dei dati tra vari formati. L'utilizzo è stato necessario per trattare tabelle contenenti i collegamenti tra nodi.
5. **R.matlab** è un pacchetto per esportare ed importare dati tra R e Matlab.
6. **Matrix** è un pacchetto utile per generare matrici ed eseguire operazioni su di esse.

## 8.1 Modello SIS deterministico

```
1 #Parametri epidemici#
2 N = 500
3 simlength <- 100
4 beta <- 0.5
5 mu <- 0.2
```

N sarà il numero di nodi, simlength la lunghezza temporale della simulazione (in giorni), beta il tasso di infettività e mu il tasso di guarigione.

Questi parametri rimarranno invariati per tutte le successive simulazioni.

Iniziamo con un'analisi classica del modello SIS studiando le quantità di infetti e suscettibili nel tempo.

```
1 #Modello SIS con eq differenziali#
2 time=seq(from=1,to=simlength,by=1)
3 initial_state_values=c(S=N-1,I=1)
4 parameters=c(mu, beta)
5
6 sis_model <- function(time,state,parameters){
7   with(as.list(c(state,parameters)),{
8
9     lambda=beta*(I/N)
10    dS=-lambda*S+mu*I
11    dI=lambda*S-mu*I
12
13    return(list(c(dS,dI)))
14  }
15 )
16 }
17
18 output<-as.data.frame(ode(y=initial_state_values,
19 func=sis_model, parms=parameters, times=time))
20 out_long=melt(output,id="time")
21 ggplot(data = out_long,
22        aes(x = time, y = value/N, colour = variable,
23            group = variable)) + geom_line() +xlab("Tempo(giorni)") +
24        ylab("inf/sus") +scale_color_discrete(name="Stato")
```

Come dalla teoria deterministica esposta nel capitolo 3, una volta definito il sistema di equazioni differenziali, riusciamo a determinarne le soluzioni e a graficarle, ottenendo la figura 3.1.

## 8.2 Epidemia su Network Omogeneo

Generiamo un grafo omogeneo seguendo il modello di Erdős-Rényi. Quest'ultimo è uno dei modelli base per i grafi aleatori: si tratta di un grafo con  $N$  nodi in cui ogni arco viene formato indipendentemente e con una probabilità  $p$  fissata. Usiamo quindi la funzione `erdos.renyi.game(N,p)` inclusa nella libreria di *iGraph* per generare il network e attraverso la funzione `as_edgelist` creiamo la lista dei collegamenti presenti che ci sarà utile negli studi successivi.

```
1 network_hom.i<-erdos.renyi.game(N, 1/25)
2 links_hom <- as_edgelist(network_hom.i)
3
4 infected_hom <- logical(N)
5 patientzero_hom <- sample(N,1)
6 infected_hom[patientzero_hom] <- TRUE
7 vettore_num_infetti_hom <- integer(simlength)
8 vettore_num_suscettibili_hom <- integer(simlength)
9
10 fixlayout <- layout.kamada.kawai(network_hom.i)
11 node.colour <- rep("SkyBlue2",N)
12 node.colour[patientzero_hom] <- "red"
13 summary(network_hom.i)
14 Max_degree_hom = (max(degree(network_hom.i)))
15 hist(degree(network_hom.i),xlim = c(0,16),
16 breaks = seq(0,Max_degree_hom,1), xaxp=c(0,16,4))
17 abline(h=seq(0,20,1), col="gray", lty="dotted" )
18 plot(network_hom.i,layout=fixlayout,
19 main="Tempo_0", vertex.color=node.colour)
```

Una volta generata la rete omogenea, si sceglie casualmente un nodo (il paziente zero) e da questo si diffonderà l'epidemia. Vediamo la struttura della rete creata e la distribuzione dei gradi dei singoli nodi.

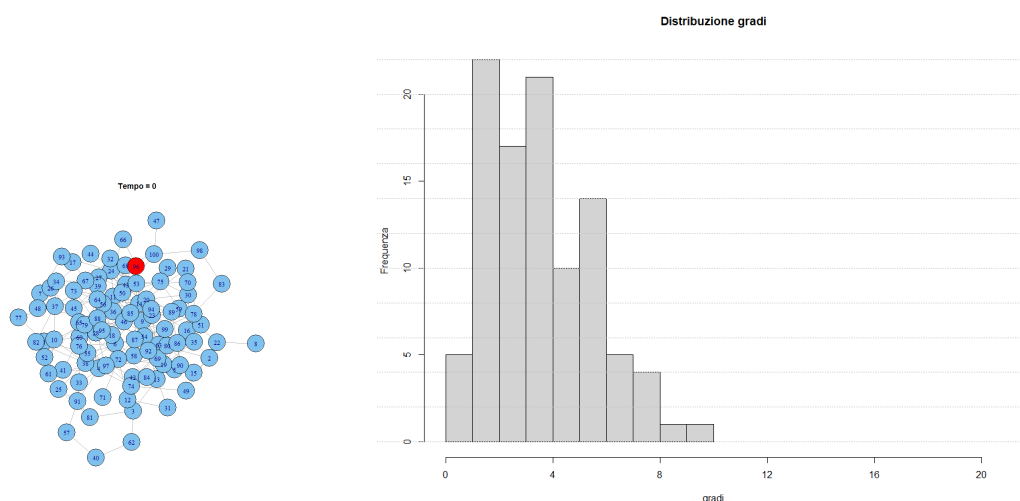


Figura 8.1: Sinistra: Network omogeneo.  
Destra: Distribuzione dei gradi dei nodi.

```

1 for (i in 1:simlength){
2   discordant.links <- which(xor(infected_hom[links_hom[,1]],
3   infected_hom[links_hom[,2]]))
4   transmit <- rbinom(length(discordant.links),1,beta)
5   transmitter.links <- discordant.links[transmit==1]
6   nodes.of.transmitter.links <- unique(as.vector
7   (links_hom[transmitter.links,1:2]))
8
9   infected_hom[nodes.of.transmitter.links] <- TRUE
10  infected.links <- which(infected_hom)
11  curable <- rbinom(length(infected.links),1,mu)
12  cured.links <- infected.links[curable==1]
13  infected_hom[cured.links] <- FALSE
14
15  vettore_num_infetti_hom[i] <- sum(infected_hom)
16  vettore_num_suscettibili_hom[i] <- N-vettore_num_infetti_hom[i]
17  print(paste0('infetti:',vettore_num_infetti_hom[i]))
18  print(paste0('suscettibili:',vettore_num_suscettibili_hom[i]))
19 }
20
21 #stampa andamento epidemico simulato#
22 plot(vettore_num_infetti_hom/N, type = 'l')
23 lines(vettore_num_suscettibili_hom/N, type = 'l', col='green')
24
25 #networks alla fine dell'evoluzione#
26 node.colour[infected] <- "red"
27 plot(network_hom.i,layout=fixlayout, vertex.color=node.colour)

```

Simuliamo l'epidemia e stampiamo i risultati. La rappresentazione dell'andamento epidemico sul grafo al termine della simulazione (ovvero passati 80 giorni) sarà la seguente (con i nodi infetti rappresentati in rosso e i suscettibili in blu):

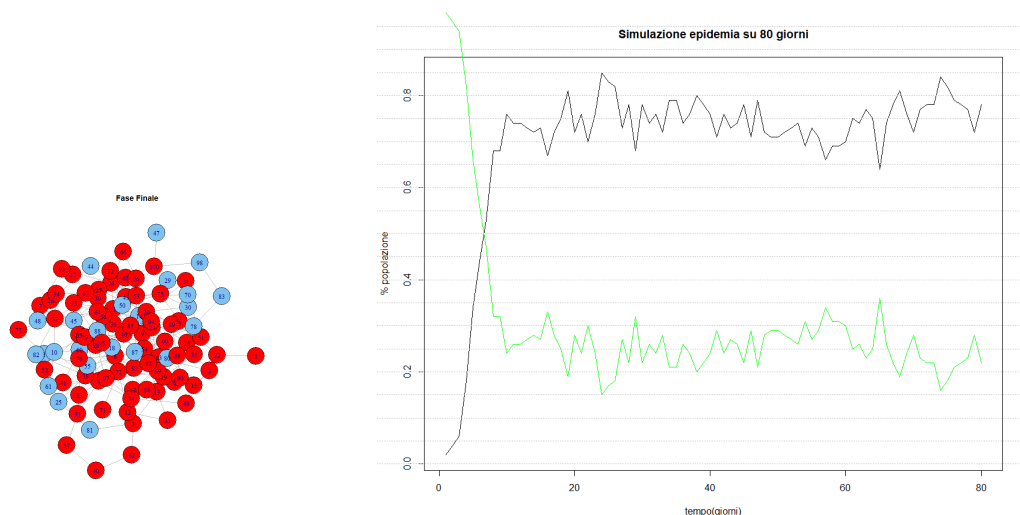


Figura 8.2: Sinistra: Grafo Omogeneo al termine delle simulazioni. Destra: Andamento epidemico (Infetti in nero e Suscettibili in verde).

## 8.3 Epidemia su Network Small-World

```

1 sm <- sample_smallworld(1, N, 2, 0.05)
2 links_sm <- as_edgelist(sm)
3
4 fixlayout <- layout.kamada.kawai(sm)
5 node.colour <- rep("SkyBlue2",N)
6 Max_degree_sm = (max(degree(sm)))
7 hist(degree(sm), xlim = c(0,16),
8 breaks = seq(0,Max_degree_sm,1), xaxp=c(0,16,4),
9 xlab="gradi", ylab = "Frequenza")
10 abline(h=seq(0,80,5), col="gray", lty="dotted" )
11 plot(sm,layout=fixlayout,main="Tempo=0",vertex.color=node.colour)

```

Generiamo ora un network di tipo *Small-World* usando la funzione `sample_smallworld` ( $dim, size, nei, p$ ) inclusa nella libreria `iGraph`. Otterremo un grafo secondo il modello Watts-Strogatz: per prima cosa viene creato un reticolo di dimensione  $dim$ , numero di nodi pari a  $size$  e un intorno per ogni nodo all'interno del quale si possono definire dei collegamenti  $nei$ . Quindi i bordi del reticolo vengono ricablati in modo uniforme e casuale con probabilità  $p$ . Vediamo la rappresentazione grafica del grafo *Small-World* generato e la rispettiva distribuzione dei nodi in 8.3.

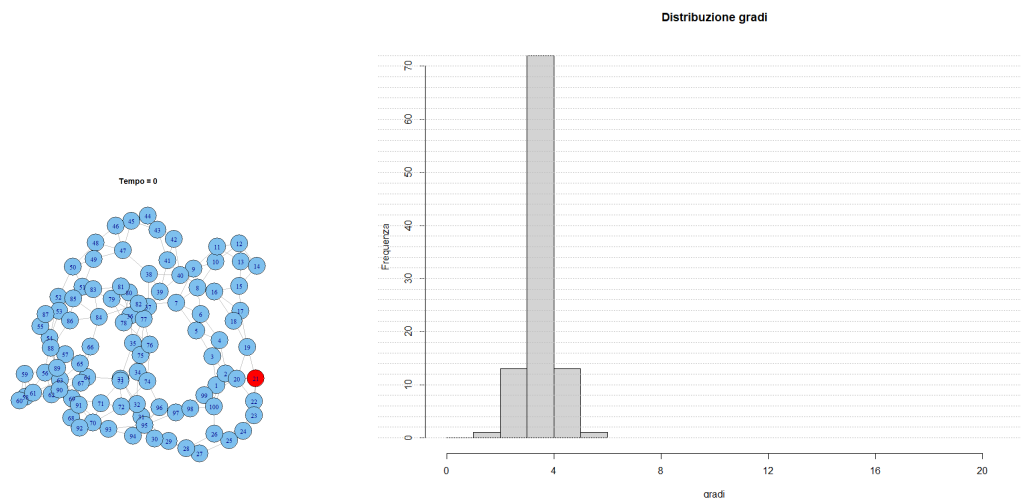


Figura 8.3: Sinistra: Grafo Small-World.  
Destra: Distribuzione dei gradi dei nodi.

```

1 infected_sm <- logical(N)
2 patientzero_sm <- sample(N,1)
3 infected_sm[patientzero_sm] <- TRUE
4
5 vettore_num_infetti_sm <- integer(simlength)
6 vettore_num_suscettibili_sm <- integer(simlength)
7
8 for (i in 1:simlength){
9   #Fase infezione#
10  discordant.links <- which(xor(

```

```

11 infected_sm[links_sm[,2]])
12 transmit <- rbinom(length(discordant.links),1,beta)
13 #aggiornamento del vettore di infezione
14 transmitter.links <- discordant.links[transmit==1]
15 nodes.of.transmitter.links <- unique(as.vector(links_sm
16 [transmitter.links,1:2]))
17 infected_sm[nodes.of.transmitter.links] <- TRUE
18
19 #Fase cura#
20 infected.links <- which(infected_sm)
21 curable <- rbinom(length(Infected.links),1,mu)
22 #aggiornamento del vettore di infezione
23 cured.links <- infected.links[curable==1]
24 infected_sm[cured.links] <- FALSE
25
26 vettore_num_infetti_sm[i] <- sum(Infected_sm)
27 vettore_num_suscettibili_sm[i] <- N-vettore_num_infetti_sm[i]
28 print(paste0('infetti:', vettore_num_infetti_sm[i]))
29 print(paste0('suscettibili:', vettore_num_suscettibili_sm[i]))
30 }
31
32 #stampa andamento epidemico simulato#
33 plot(vettore_num_infetti_sm/N, type = 'l', xlab = 'tempo',
34 ylab = '%popolazione')
35 lines(vettore_num_suscettibili_sm/N, type = 'l', col='green')

```

Simuliamo l'andamento epidemico e stampiamo i risultati grafici ottenuti al termine della simulazione.

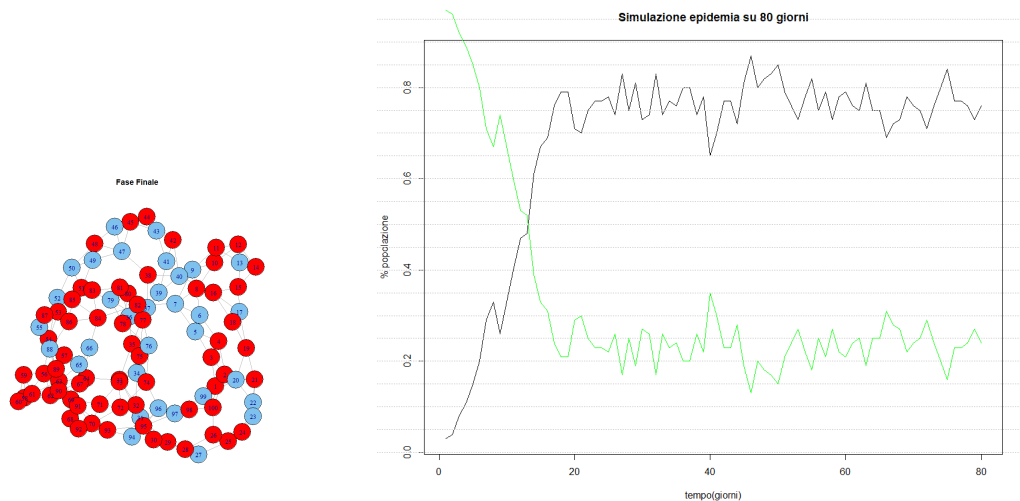


Figura 8.4: Sinistra: Grafo Small-World al termine delle simulazioni (t=80 giorni).  
 Destra: Andamento epidemico (Infetti in nero e Suscettibili in verde).

## 8.4 Epidemia su Network Scale-Free

```
1 #funzione per generare networks di tipo Barabasi#
2 generate.network.B <- function(N,links.per.step){
3   L <- matrix(nrow=0,ncol=2)
4   deg <- integer(N)
5   for (i in 2:N) {
6     n.new <- min(links.per.step,i-1)
7     linkto <- sample(i-1,n.new,prob=deg[1:(i-1)]+1)
8     newlinks <- cbind(rep(i,n.new),linkto)
9     L <- rbind(L,newlinks)
10    deg[i] = deg[i] + n.new
11    deg[linkto] = deg[linkto]+1
12  }
13  colnames(L) <- NULL
14  L
15 }
```

Generiamo un network di tipo Barabasi che si costruisce partendo da un singolo nodo e aggiungendo un nodo alla volta. Ogni volta che viene aggiunto un nuovo nodo, viene connesso ad alcuni dei vecchi in modo casuale. La probabilità che i vecchi nodi ricevano un nuovo collegamento aumenta con il loro grado. Man mano che il grafico cresce, la distribuzione dei gradi dei nodi si avvicinano ad una power law della forma  $P(k) \sim k^{-\gamma}$ . Tali reti sono anche chiamate *scale-free*.

```
1 #Modello SIS simulato su network di Barabasi#
2 plot.spread <- TRUE
3 links <- generate.network.B(N,2)
4 infected <- logical(N)
5 patientzero <- sample(N,1)
6 infected[patientzero] <- TRUE
7 vettore_num_infetti <- integer(simlength)
8 vettore_num_suscettibili <- integer(simlength)
9
10 network.i <- graph.edgelist(links,directed=FALSE)
11 fixlayout <- layout.kamada.kawai(network.i)
12 node.colour <- rep("SkyBlue2",N)
13 node.colour[patientzero] <- "red"
14 Max_degree = (max(degree(network.i)))
15 hist(degree(network.i),xlim = c(0,25),
16 breaks = seq(0,Max_degree,1), xaxp=c(0,25,5),
17 xlab="gradi", ylab = "Frequenza")
18 abline(h=seq(0,40,2), col="gray", lty="dotted" )
19 plot(network.i,layout=fixlayout, main="Tempo=0",
20 vertex.color=node.colour)
```

In 8.5 possiamo vedere la distribuzione dei gradi dei nodi e il network generato.

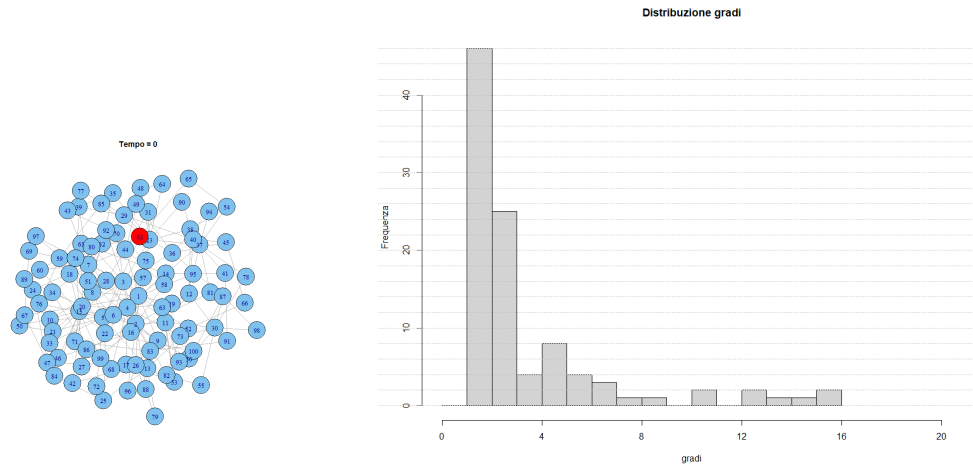


Figura 8.5: Sinistra: Network Scale free.  
Destra: Distribuzione dei gradi dei nodi.

```

1 for (i in 1:simlength){
2   #Fase infezione#
3   discordant.links <- which(xor(infected[links[,1]],
4     infected[links[,2]]))
5   transmit <- rbinom(length(discordant.links),1,beta)
6   #aggiornamento del vettore di infezione
7   transmitter.links <- discordant.links[transmit==1]
8   nodes.of.transmitter.links <- unique(as.vector
9     (links[transmitter.links,1:2]))
10  infected[nodes.of.transmitter.links] <- TRUE
11
12  #Fase cura#
13  infected.links <- which(infected)
14  curable <- rbinom(length(infected.links),1,mu)
15  cured.links <- infected.links[curable==1]
16  infected[cured.links] <- FALSE
17
18  vettore_num_infetti[i] <- sum(infected)
19  vettore_num_suscettibili[i] <- N-vettore_num_infetti[i]
20  print(paste0('infetti:', vettore_num_infetti[i]))
21  print(paste0('suscettibili:', vettore_num_suscettibili[i]))
22
23  #Stampa grafo con infetti e suscettibili#
24  if (plot.spread){
25    node.colour[infected] <- "red"
26    plot(network.i,layout=fixlayout, main=paste("Tempo=", i),
27      vertex.color=node.colour)
28  }
29 }
30
31 #networks alla fine dell'evoluzione#
32 node.colour[infected] <- "red"
33 plot(network.i, layout=fixlayout, main=paste("Momento_attuale"),
34   vertex.color=node.colour)

```



```

35
36 #stampa andamento epidemico simulato#
37 plot(vettore_num_infetti/N, type = 'l',
38 main = 'Simulazione_epidemia', xlab = 'tempo(giorni)',
39 ylab = '%popolazione')
40 lines(vettore_num_suscettibili/N, type = 'l', col='green')

```

Simuliamo l'andamento epidemico e stampiamo i risultati.

L'andamento epidemico finale su grafico è rappresentato in figura: 8.6:

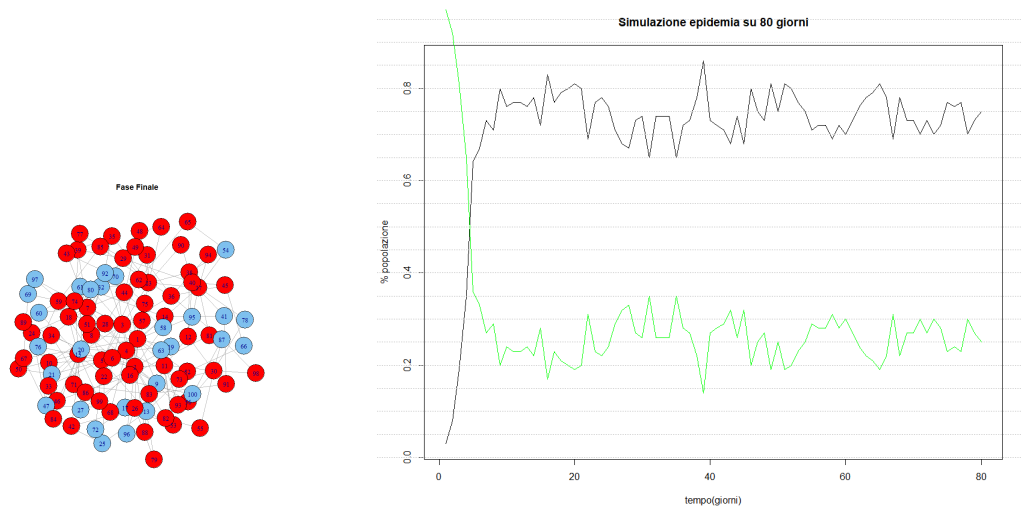


Figura 8.6: Sinistra: Network Scale free dopo 80 giorni dall'inizio dell'epidemia.  
 Destra: Andamento epidemia (Infetti in nero e Suscettibili in verde).

## 9.1 Andamenti

Rappresentiamo gli andamenti simulati delle epidemie sui tre diversi networks e confrontiamoli con il modello deterministico classico.

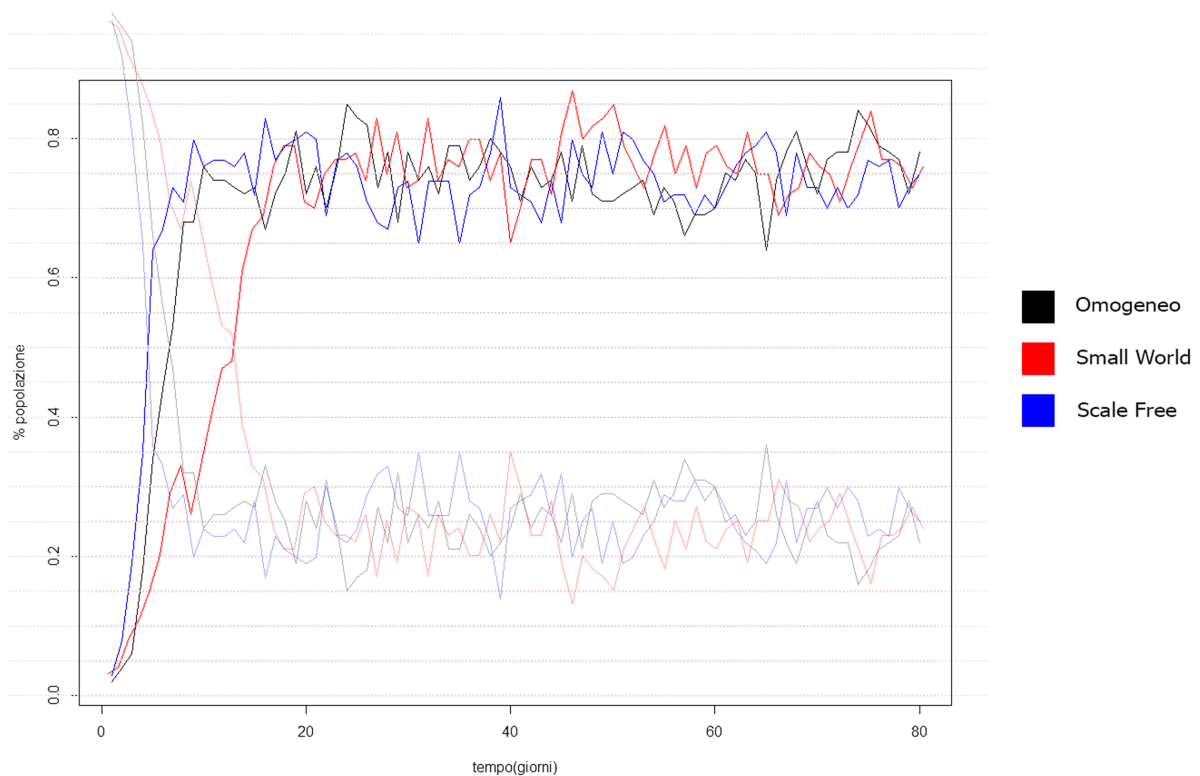


Figura 9.1: Confronto tra tutte le epidemie

Dalla figura (9.1) possiamo vedere come solo l'epidemia sul grafo small world inizialmente segua lontanamente l'andamento del modello deterministico. Dopo dodici giorni invece anche quest'ultima assume una crescita più accentuata fino a raggiungere i livelli

delle altre due epidemie.

Per un confronto più dettagliato, mostriamo come varia il numero di infetti nelle 3 diverse epidemie per 300 giorni dall'infezione del paziente-zero.

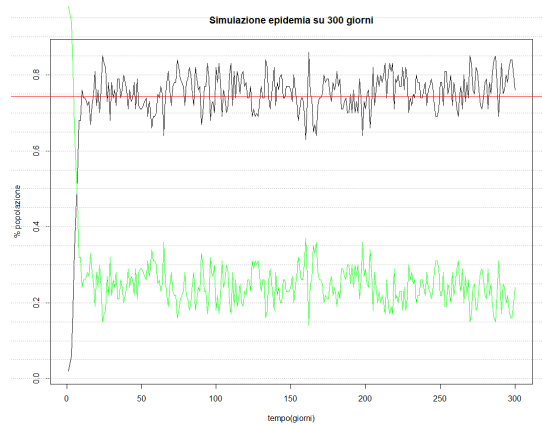


Figura 9.2: Epidemia su Network Omogeneo

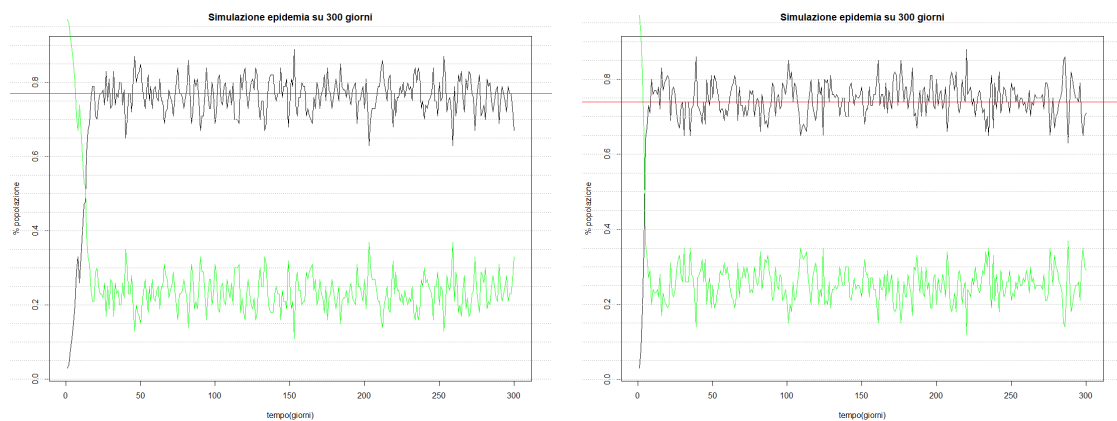


Figura 9.3: Sinistra: Epidemia su Small World. Destra: Epidemia su Scale Free.

Tutte e tre le epidemie hanno un numero di infetti (e quindi anche suscettibili) che fluttua intorno all'equilibrio endemico descritto nel capitolo (3.4.1) e rappresentato dalle rette orizzontali rosse in (9.2) e (9.3). Tuttavia, diversamente dal capitolo (3.4.1) il livello endemico assunto dalle tre è notevolmente più elevato di quello previsto dal modello deterministico che, dati i parametri epidemici, ha il valore di  $3/5$ .

I livelli endemici delle epidemie simulate sono stati calcolati eseguendo un fit lineare dei vettori contenenti il numero di suscettibili ed infetti attraverso la funzione `lm(formula)`, inclusa in R. Più nel dettaglio, partendo dal vettore la cui *i-esima* componente contiene il numero di infetti al tempo *i*, sono stati eliminati i dati dei primi giorni di infezione (momento di non equilibrio) per poi calcolare un fit lineare sulle componenti rimaste.

Vediamo la sezione del codice in R per il calcolo:

```
1 time_n=seq(from=25,to=simlength,by=1)
2 vettore_num_infetti_n=vettore_num_infetti[25:simlength]
3 linear_model<-lm(vettore_num_infetti_n/N ~ time_n)
4 abline(h=as.numeric(unlist(linear_model[[1]][1])), col='red')
```

Il risultato è stata una retta con pendenza nulla (come ci si poteva aspettare dal grafico) e con intercetta pari al livello endemico. I risultati per ogni epidemia e per il numero di infetti sono illustrati qui sotto (normalizzati sul numero totale di individui):

- Epidemia su network omogeneo: livello endemico = 0.7440
- Epidemia su network Small World: livello endemico = 0.7700
- Epidemia su network Scale Free: livello endemico = 0.7387

Per avere una visione più chiara dell'andamento dei livelli endemici, simuliamo 100 epidemie su ciascun grafo e calcoliamone i rispettivi livelli endemici. I risultati saranno tre vettori (uno per ogni grafo) di 100 componenti ciascuno: nella posizione  $i$  c'è il livello endemico dell' $i$ -esima epidemia. Vediamo un *summary* dei tre vettori:

- Vettore endemico del grafo omogeneo:

| Min.   | Media  | Max.   | Var.  |
|--------|--------|--------|-------|
| 0.0000 | 0.7003 | 0.7699 | 0.037 |

- Vettore endemico del grafo Small World:

| Min.   | Media  | Max.   | Var.  |
|--------|--------|--------|-------|
| 0.0000 | 0.7088 | 0.7873 | 0.044 |

- Vettore endemico del grafo Scale Free:

| Min.   | Media  | Max.   | Var.  |
|--------|--------|--------|-------|
| 0.0000 | 0.7170 | 0.7723 | 0.022 |

Notiamo la presenza, in tutti e tre i vettori, di almeno un livello endemico nullo che corrisponde al caso in cui l'epidemia si estingue arrivando ad un numero di infetti pari a zero.

## 9.2 Risoluzione di (7.4) nel caso stazionario

Come promesso nel capitolo (7.2), risolviamo l'equazione (7.4) applicata a ciascun grafo, precedentemente analizzato, nel caso stazionario, così da poter determinare i punti di equilibrio della probabilità che il nodo  $i$  sia infetto al tempo  $t$  per ogni nodo nel grafo.

Per fare quanto detto, si dovranno determinare gli zeri di un sistema con  $N$  equazioni e  $N$  variabili, dove  $N$  è il numero di nodi presenti nella rete. Il nostro obiettivo è quello di determinare i punti di equilibrio per grafi con quanti più nodi possibile. Dovremo quindi scegliere un metodo di risoluzione per il sistema (7.4) tale che sia, computazionalmente, il più efficiente possibile. Abbiamo già a disposizione lo script di MatLab, usato nel capitolo (7.2), che sfrutta il metodo di Newton. Tuttavia questo metodo prevede l'utilizzo della matrice inversa dello Jacobiano che, per sistemi a dimensioni particolarmente elevate, risulta molto lento (implementandolo si riesce a risolvere il sistema per  $N \leq 5$ ).

Un metodo computazionalmente più vantaggioso è quello del *Gradient Descent*: è un algoritmo di risoluzione per problemi di ottimizzazione non vincolati. Si parte dalla funzione  $f(x)$  a più variabili che vogliamo risolvere, ne determiniamo lo jacobiano ed eseguiamo delle iterazioni in cui, ad ogni passo  $k$ , supposto  $\nabla f(x_k)$  diverso da zero, si sceglie, come direzione di ricerca, la direzione  $d_k = -\nabla f(x_k)$ . Così facendo gli  $x_k$  si avvicineranno sempre di più al punto di minimo (e quindi equilibrio stabile) della funzione. Il metodo teorico usato è trattato in dettaglio nell'appendice B.

Usando le stesse notazioni del capitolo (7.2), partiamo dal sistema (7.4) e ne determiniamo il caso stazionario ponendo le singole equazioni uguali a zero:

$$-\rho_i^I(t) + \lambda[1 - \rho_i^I(t)] \sum_{j=1}^N a_{ij} \rho_j^I(t) = 0 \quad (9.1)$$

Definiamo ora la funzione  $F(x)$  come somma dei quadrati delle singole componenti di (9.1) ottenendo così:

$$F(x) = \sum_{i=1}^N \left[ -\rho_i^I(t) + \lambda[1 - \rho_i^I(t)] \sum_{j=1}^N a_{ij} \rho_j^I(t) \right]^2 \quad (9.2)$$

$F(x)$  sarà la funzione che minimizzeremo attraverso il *Gradient Descent*. Notiamo inoltre che possiamo esprimere  $F(x)$  nel seguente modo:

$$F(\mathbf{x}) = \frac{1}{2} f(\mathbf{x})^T \cdot f(\mathbf{x})$$

con  $f(\mathbf{x}) = \beta \cdot \text{diag}(\mathbf{1} - \mathbf{x})\mathbf{A}\mathbf{x} - \mu\mathbf{x}$ , relazione che ci permette di definire  $F(\mathbf{x})$  in MatLab senza dover eseguire particolari operazioni.

Prendiamo dunque i 3 grafi (Omogeneo, Small-World e Scale Free) su cui avevamo simulato le epidemie, ne determiniamo le matrici di connettività (necessarie per definire le equazioni (9.1) per ciascun grafo), che a questo punto saranno matrici 100x100, e le importiamo nello script di MatLab.

```
1 A_hom = cell2mat(struct2cell(load('A_hom.mat')));
2 sol_hom = Solver(A_hom);
3
4 A_sm = cell2mat(struct2cell(load('A_sm.mat')));
5 sol_sm = Solver(A_sm);
6
7 A_sf = cell2mat(struct2cell(load('A_sf.mat')));
8 sol_sf = Solver(A_sf);
```

Richiamiamo ora, per ogni matrice di connettività, la funzione *Solver*, che ha la seguente struttura:

```
1 function sol = Solver(A)
2
3 % Parameters
4     Beta = 0.5;
5     Mu = 0.2;
6     [N, ~] = size(A);
7
8 % Variables
```

```

9      syms X [1 N];
10     X = X.';
11
12 % Functions definition
13     f = simplify(Beta*diag(1-X)*A*X-Mu*X);
14
15 % Jacobian of f
16     J = simplify(jacobian(f, X));
17
18 x0 = 0.6*ones(N, 1);
19 [sol,its,err] = Gradient_descent(N, f, J, x0);
20 sol = round(vpa(sol), 4);
21
22 disp('iterations:')
23 disp(its)
24 disp('error:')
25 disp(vpa(err))
26 disp('zeroes_values:')
27 disp(sol)
28 end

```

I parametri usati nella definizione del problema sono gli stessi delle simulazioni precedenti, la funzione è definita in modo vettoriale secondo l'equazione (7.6) e il calcolo dello jacobiano è stato eseguito attraverso la funzione *jacobian* (disponibile in MatLab senza alcun tipo di libreria). Si è poi usata la funzione *simplify* per eseguire semplificazioni algebriche delle funzioni trovate. Una volta definito il vettore iniziale  $x_0$  da cui far partire le iterazioni, si richiama la funzione *Gradient Descent* che va ad eseguire effettivamente i calcoli avendo come input la dimensione  $N$  del problema, la funzione  $f$ , il suo jacobiano e  $x_0$ .

```

1 function [Xi,its,err] = Gradient_descent(N, F, jacF, Xi)
2
3     syms X [1 N];
4     X = X.';
5
6     tol = 10^-6;
7     err = 1;
8     maxits = 100;
9     its = 0;
10
11     jacF_t = jacF.';
12     Xold_old = zeros(N,1);
13
14     while (err>tol) && (its<maxits)
15         Xold = Xi;
16         Fn_Fn1 = (((vpa(subs(jacF_t, X, Xold))))*
17             (vpa(subs(F, X, Xold))))-((vpa(subs(jacF_t, X,
18             Xold_old))))*(vpa(subs(F, X, Xold_old))));
19         gamma_i = vpa(abs((Xold-Xold_old).'*Fn_Fn1)
20             /(norm(Fn_Fn1)^2));
21         Xi = Xold - gamma_i * (vpa(subs(jacF_t, X, Xold)))*
22             (vpa(subs(F, X, Xold)));

```

```

23     err = norm(Xi-Xold);
24     Xold_old = Xold;
25     its = its+1;
26     disp(its);
27 end
28 end

```

Si può notare la presenza di un fattore moltiplicativo  $\gamma_i$ : quest'ultimo influisce sulla distanza tra l'iterata precedente e la successiva, per cui, una sua errata valutazione, porterebbe ad un fallimento del metodo (potrebbe essere troppo piccolo o troppo grande, causandone la non convergenza). Esistono diversi modi per determinarne il valore corretto. Personalmente ho usato il metodo *Barzilai–Borwein*, anche questo approfondito nell'appendice B.

Eseguendo il codice, otteniamo i punti di equilibrio per ciascun grafo. I risultati verranno presentati graficamente su un piano cartesiano in cui l'asse  $x$  corrisponde ai gradi dei singoli nodi e l'asse  $y$  al rispettivo punto di equilibrio (infatti il valore del punto di equilibrio della probabilità che il nodo  $i$  sia infetto al tempo  $t$  dipende esclusivamente dal grado del nodo stesso).

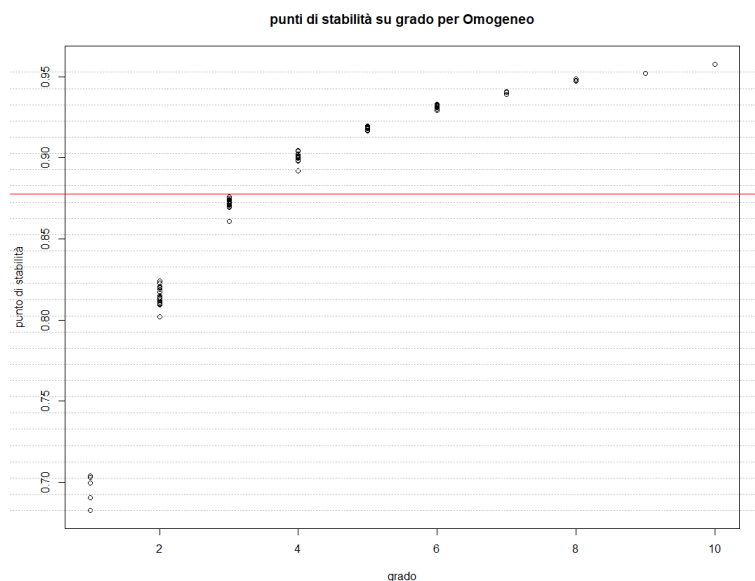
- Grafo omogeneo:

Iterazioni eseguite: 74

Errore:  $0,96 \times 10^{-6}$

Valor medio del punto di stabilità (retta rossa nel grafico): 0.8778

Punto di stabilità minimo: 0.683    Punto di stabilità massimo: 0.958



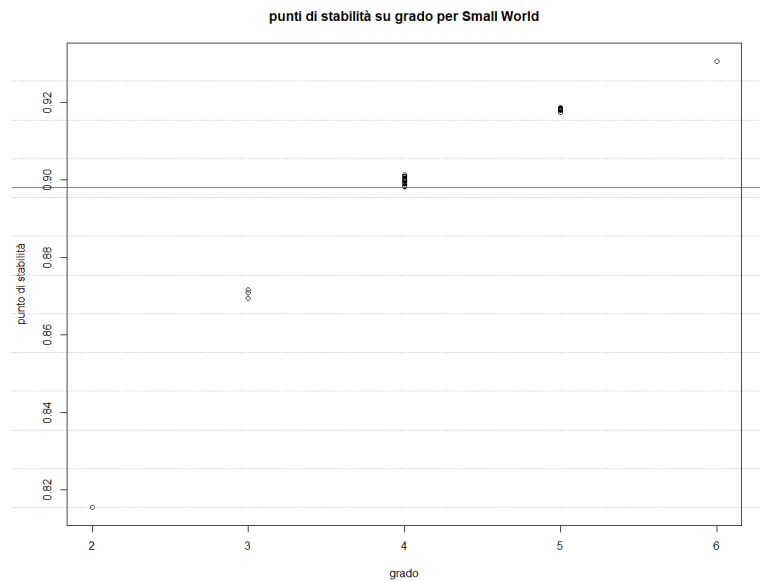
- Grafo Small World:

Iterazioni eseguite: 28

Errore:  $0,77 \times 10^{-6}$

Valor medio del punto di stabilità (retta rossa nel grafico): 0.8981

Punto di stabilità minimo: 0.816    Punto di stabilità massimo: 0.931



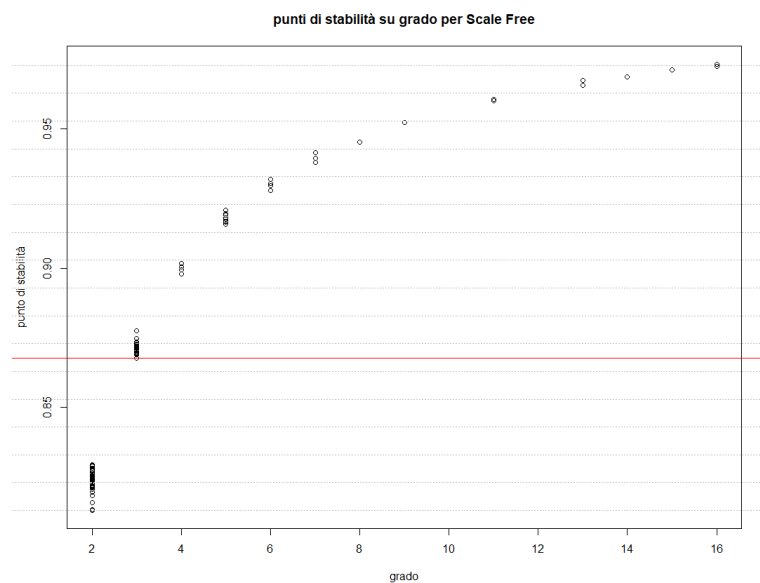
- Grafo Scale Free:

Iterazioni eseguite: 71

Errore:  $0,88 \times 10^{-6}$

Valor medio del punto di stabilità (retta rossa nel grafico): 0.8677

Punto di stabilità minimo: 0.813    Punto di stabilità massimo: 0.973





## APPENDICE A

---

### Tempo all'estinzione epidemica

---

Prima di enunciare il teorema, abbiamo bisogno di una definizione. Diciamo che  $a_n$  e  $b_n$  sono *asintoticamente equivalenti* per  $n \rightarrow \infty$  (matematicamente lo indicheremo con  $a_n \sim b_n$ ), se il quoziente  $a_n/b_n$  tende all'unità quando  $n$  diventa grande.

**Teorema 2.** *Il tempo  $T_n$  all'estinzione del modello SIS stocastico ha le seguenti proprietà asintotiche:*

1.  $\beta > \mu$  e  $m_n/n \rightarrow \gamma > 0$  per  $n \rightarrow \infty \Rightarrow T_n/E(T_n) \rightarrow U$  in distribuzione, dove  $U$  ha distribuzione esponenziale con parametro 1. In più

$$E(T_n) \sim \sqrt{\frac{2\pi}{n}} \frac{\beta}{(\beta - \mu)^2} e^{nV} \quad (\text{A.1})$$

per  $n \rightarrow \infty$ , dove  $V = \log(\beta/\mu) - 1 + \mu/\beta > 0$

2.  $\beta > \mu$  e  $m_n = m \geq 1$  per ogni  $n \Rightarrow T_n \rightarrow T$  dove  $\mathbf{P}(T < \infty) = (\mu/\beta)^m < 1$ . In questo caso  $T$  è il tempo di estinzione per un processo lineare di nascita e morte con tasso di natalità  $\beta$ , tasso di mortalità  $\mu$  e con  $m$  individui infetti all'istante iniziale. Nell'insieme dove  $T$  è infinito,  $T_n/E(T_n) \rightarrow U$  in distribuzione, con  $U$  e  $E(T_n)$  come sopra.

3.  $\beta \leq \mu$  e  $m_n/n \rightarrow \gamma > 0$  per  $n \rightarrow \infty \Rightarrow$  abbiamo che

$$(\mu - \beta(1 - \gamma))T_n - \log n - \log \gamma - \log 1 - \frac{\beta(1 - \gamma)}{\mu} \rightarrow W$$

in distribuzione, dove  $W$  ha la distribuzione del valore estremo:

$$\mathbf{P}(W \leq w) = \exp\{-e^{-w}\}$$

4.  $\beta \leq \mu$  e  $m_n = m \geq 1$  per ogni  $n \Rightarrow T_n \rightarrow T$  con  $T$  come in 3, ma ora  $\mathbf{P}(T < \infty) = 1$

## APPENDICE B

---

### Metodo del Gradient Descent

---

Il *Gradient Descent* è un algoritmo di ottimizzazione iterativo del primo ordine per trovare un minimo locale di una funzione differenziabile.

Se la funzione a più variabili  $F(x)$  è ben definita e differenziabile nell'intorno di un punto  $a$ , allora  $F(x)$  diminuisce più velocemente se si va da  $a$  nella direzione del gradiente negativo di  $F$ . Segue che, se

$$a_{n+1} = a_n - \gamma \nabla F(a_n)$$

per un passo  $\gamma \in \mathbb{R}_+$  sufficientemente piccolo, allora  $F(a_n) \geq F(a_{n+1})$ . In altre parole, il termine  $\gamma \nabla F(a)$  è sottratto da  $a$  perché ci vogliamo muovere contro il gradiente, verso il minimo locale. Si inizia quindi da un  $x_0$  per poi costruire una sequenza  $x_0, x_1, x_2, \dots$  così definita:

$$x_{n+1} = x_n - \gamma_n \nabla F(x_n), \quad n \geq 0$$

Si crea così una sequenza monotona  $F(x_0) \geq F(x_1) \geq F(x_2) \geq \dots$  e la sequenza  $x_0, x_1, x_2, \dots$  converge al minimo locale. Si noti che il valore del passo  $\gamma$  può cambiare ad ogni iterazione. Sotto opportune ipotesi della funzione  $F$  (convessa e  $\nabla F$  di Lipschitz) e per un'opportuna scelta di  $\gamma$  (attraverso una *line search* o il metodo di *Brezili-Borwein* mostrato qui sotto),

$$\gamma_n = \frac{|(x_n - x_{n-1})^T [\nabla F(x_n) - \nabla F(x_{n-1})]|}{\|\nabla F(x_n) - \nabla F(x_{n-1})\|^2}$$

la convergenza ad un minimo locale è garantita. Quando la funzione  $F$  è convessa, tutti i minimi locali sono globali, allora in questo caso il metodo del *gradient descent* porta alla soluzione globale.

---

## Bibliografia

---

- [1] Andersson H., Britton T.; *Stochastic epidemic models and their statistical analysis.*
- [2] Chakrabarti, Wang, Leskovec, Faloutsos; *Epidemic thresholds in real networks.*
- [3] Easley D., Kleinberg J.; *Networks, crowds, and markets: Reasoning about a highly connected world.*
- [4] Pastor-Satorras, Castellano, Van Mieghem, Vespignani; *Epidemic processes in complex networks.*
- [5] Lajmanovich A., Yorke J.; *A Deterministic Model for Gonorrhea in a Nonhomogeneous Population.*
- [6] Doshi V., Mallick S., Do Young Eun; *Competing Epidemics on Graphs - Global Convergence and Coexistence.*
- [7] M. Grasmair; *Newton's Method in Higher Dimensions.*