

Studio dell'evoluzione di un agente patogeno tramite lo sviluppo di un programma in C++

Belardinelli Giulia, Ronchetti Filippo

17 luglio 2023

Sommario

In questa relazione siamo andati a studiare il comportamento di un'epidemia attraverso la progettazione di un codice in linguaggio C++ che ne riproducesse la proliferazione secondo il modello SIR. I risultati riproducono l'andamento atteso al variare dei parametri inseriti. In particolare sono riportati i risultati per le coppie di valori $(\beta = 0.5, \gamma = 0.5)$, $(\beta = 0.7, \gamma = 0.2)$ e $(\beta = 0.2, \gamma = 0.7)$ su una popolazione ed un I_0 fissati.

1 Introduzione

Molto spesso i modelli matematici sono utilizzati in *epidemiologia* per studiare la nascita e l'evoluzione di un determinato fenomeno di interesse sanitario, ad esempio il diffondersi di una malattia. Alla base della biomatematica, questi modelli danno una descrizione deterministica dell'evoluzione temporale dell'epidemia grazie all'utilizzo di una o più equazioni (con annesse condizioni al contorno).

1.1 Il modello SIR

Il modello SIR si basa sull'evoluzione di un'epidemia all'interno di una popolazione chiusa N , divisa, come si può intuire dal nome, in tre compartimenti distinti:

1. S = Suscettibili, ovvero le persone che possono essere infettate;
2. I = Infetti, ovvero le persone contagiate e che successivamente diffondono l'infezione;
3. R = Rimossi, ovvero le persone guarite (o morte) che non possono più né infettarsi né infettare.

Lo stato di una persona può variare tra S, I ed R solamente in una direzione, ovvero prima $S \rightarrow I$ e successivamente $I \rightarrow R$. Essendo una popolazione chiusa dovranno valere le seguenti espressioni

$$S, I, R \in N \quad (1a)$$

$$S(t) + I(t) + R(t) = N \quad (1b)$$

L'evoluzione della malattia all'interno di questo modello è descritta dalle seguenti equazioni differenziali

$$\frac{dS}{dt} = -\beta \frac{S}{N} I \quad (2a)$$

$$\frac{dI}{dt} = \beta \frac{S}{N} I - \gamma I \quad (2b)$$

$$\frac{dR}{dt} = \gamma I \quad (2c)$$

dove $\beta \in [0, 1]$ è la misura della probabilità di contagio in seguito a contatti tra infetti e suscettibili, mentre $\gamma \in [0, 1]$ è la probabilità di guarigione (o morte) di un infetto, ed è l'inverso del tempo medio di risoluzione dell'infezione. Possiamo notare che l'epidemia sarà in espansione quando $dI/dt > 0$, ovvero quando $S/N > \beta/\gamma$. Una volta raggiunto il valore I_{max} , l'epidemia sarà in regressione ed il numero di infetti comincerà a calare.

Discretizzando le equazioni differenziali precedentemente citate e posto $\Delta t = 1$, otterremo le seguenti espressioni

$$S_i = S_{i-1} - \beta \frac{S_{i-1}}{N} I_{i-1} \quad (3a)$$

$$R_i = R_{i-1} + \gamma I_{i-1} \quad (3b)$$

$$I_i = I_{i-1} + \beta \frac{S_{i-1}}{N} I_{i-1} - \gamma I_{i-1} \quad (3c)$$

Bisogna infine tener conto che le malattie studiate tramite questo modello hanno i seguenti limiti:

- ogni individuo può essere infettato da un agente patogeno una sola volta;
- la popolazione è costante, per cui le nascite e morti per altre ragioni non sono prese in considerazione;
- gli infetti sono immediatamente contagiosi, mentre nella realtà è spesso presente un periodo di incubazione;
- i parametri β e γ nel modello sono costanti, nella realtà sono tipicamente variabili e dipendono sia dai fattori esterni in cui si evolve la malattia, sia dalle misure di mitigazione presenti (ad esempio quarantene, vaccini...).

1.2 Obiettivi

I modelli matematici sono di rilevante importanza in epidemiologia. Un buon modello permette di simulare ciò che potrà accadere in natura e può quindi rappresentare un grande alleato nello studio delle malattie e nella creazione di misure preventive e di contenimento (isolamento, mascherine...). L'obiettivo di questa ricerca consiste infatti nell'elaborazione di un programma che simuli l'evoluzione di un virus all'interno di una popolazione chiusa, così da poterne prevedere la diffusione in un determinato arco di tempo.

2 Lo sviluppo del progetto

2.1 Main

Il corpo del codice contiene i parametri richiesti all'utente come standard input tramite i comandi *cout* e *cin*. I parametri sono i seguenti:

1. nome della malattia (*name*);
2. probabilità di contagio β (*BETA*);
3. probabilità di guarigione γ (*GAMMA*);
4. campione di popolazione (*N*);

5. numero iniziale di infetti (I_0);
6. giorni di sviluppo dell'epidemia (*days*).

A seguire viene creato l'oggetto *d* della classe *Disease* chiamando il costruttore: qui vengono utilizzati i valori precedentemente elencati da 1 a 5. Successivamente viene chiamato il metodo *evolve* sull'oggetto *d*, il quale simula l'evoluzione dell'epidemia per il numero di giorni scelto dall'utente.

Infine la funzione *f_print()* scrive sul file di testo *epidemic.txt* il risultato della funzione *print()*.

2.2 Disease

La classe *Disease* comprende al suo interno gli attributi (privati) ed i metodi (pubblici) riportati di seguito.

2.2.1 Attributi

```

1  std::string name_; //nome della malattia
2  std::vector<State> state_; //vettore degli stati
3  double tot_; //popolazione esposta
4  double gamma_; //parametro GAMMA
5  double beta_; //parametro BETA/N

```

2.2.2 Metodi

```

1  Disease(std::string p, int n, double inf, double b, double y);
2  //creazione di oggetti ed inizializzazione degli attributi
3
4  State evolve_(State const& begin);
5  //restituzione dello stato i calcolato attraverso lo stato i-1
6
7  void evolve(int n);
8  //creazione degli stati giorno dopo giorno
9
10 void print();
11 //generazione del contenuto della stampa
12
13 void f_print();
14 //inserimento del contenuto generato da print() su file

```

Nel file di implementazione sono descritte le condizioni per evoluzione degli stati. La linea

```

1  State disease::Disease::evolve_(State const &begin) {}

```

riproduce lo stato grazie all'inserimento dello stato precedente attraverso le equazioni Eq.3 e la condizione $N = s + i + r$, che descrivono l'evoluzione della malattia nel modello.

Successivamente l'array *state_* aggiunge volta per volta il nuovo stato costruito su quello precedente.

```

1 void disease::Disease::evolve(int n) {
2   for (int i = 0; i < n; ++i) {
3     state_.push_back(evolve_(state_[i]));
4   }
5 }

```

Infine viene costruito il contenuto di stampa, nel quale si è deciso di inserire:

- nome della malattia;
- valore $R_0 = \beta/\gamma$;
- valori giornalieri di Suscettibili, Infetti e Rimossi (riproducendo per il totale dei giorni il vettore degli stati, precedentemente riempito grazie ad *evolve*).

Tutto ciò, attraverso *f_print*, viene inserito all'interno di un file di testo.

2.3 Esecuzione

Per l'esecuzione andiamo a scrivere direttamente dal terminale:

```
1 //compilazione
2 g++ -c main.cpp disease.cpp
3 g++ -o sir main.o disease.o
4
5 //running
6 ./sir
```

Dopo aver inserito i parametri elencati nella Sezione 2.1, i risultati saranno usufruibili attraverso il file *epidemic.txt*. Per visualizzare i risultati direttamente da terminale digitare

```
1 cat epidemic.txt
```

Di seguito proponiamo un esempio di esecuzione, il quale riproduce valori per $\beta = 0.5$, $\gamma = 0.5$, $N = 22500$, $I_0 = 5$, days= 300 ed i cui valori sono riportati in Fig.3.

```
root@DESKTOP-9USME90:~/progetto-SIR# g++ -c main.cpp disease.cpp
root@DESKTOP-9USME90:~/progetto-SIR# g++ -o sir main.o disease.o
root@DESKTOP-9USME90:~/progetto-SIR# ./sir
Welcome to SIR simulator!
Please insert the name of the disease: Covid
Please insert Beta value: 0.5
Please insert Gamma value: 0.5
Please insert the population sample: 22500
Please insert the initial number of infected people: 5
Please insert the number of days of development of the epidemic: 300
Operation completed: file epidemic.txt is ready!
root@DESKTOP-9USME90:~/progetto-SIR# cat epidemic.txt
Disease name: Covid
Current value of R0: 1
```

Day	Susceptible	Infectuos	Recovered
0	22495	5	0
1	22492	4	2
2	22490	4	4
3	22487	4	7
4	22485	4	9
5	22482	4	12
6	22480	4	14
7	22477	4	17
8	22475	4	19
9	22472	4	22
10	22470	4	24
11	22467	4	27
12	22465	4	29

Figura 1: Esempio di compilazione da terminale.

Il codice viene eseguito correttamente, senza la presenza di Errori e Warning.

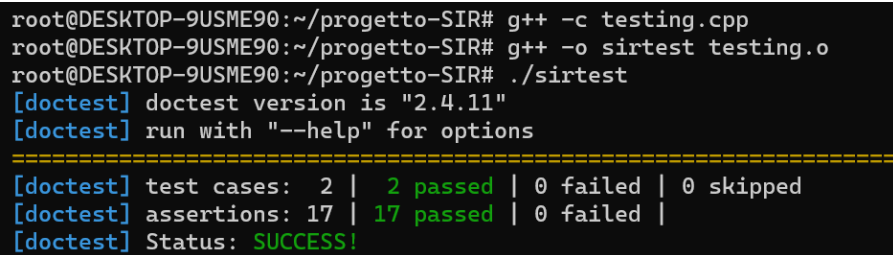
2.3.1 Testing

Come ultima prestazione abbiamo voluto inserire anche un test di verifica, al cui interno sono stati inseriti alcuni valori di prova.

Per l'esecuzione andiamo a scrivere direttamente dal terminale:

```
1 //compilazione
2 g++ -c testing.cpp
3 g++ -o sirtest testing.o
4
5 //running
6 ./sirtest
```

Anche in questo caso è riportata l'esecuzione da terminale



```
root@DESKTOP-9USME90:~/progetto-SIR# g++ -c testing.cpp
root@DESKTOP-9USME90:~/progetto-SIR# g++ -o sirtest testing.o
root@DESKTOP-9USME90:~/progetto-SIR# ./sirtest
[doctest] doctest version is "2.4.11"
[doctest] run with "--help" for options
=====
[doctest] test cases:  2 |  2 passed | 0 failed | 0 skipped
[doctest] assertions: 17 | 17 passed | 0 failed |
[doctest] Status: SUCCESS!
```

Figura 2: Esempio di compilazione da terminale

E' possibile vedere che non sono stati generati errori e che il test ha avuto successo. Nel prossimo capitolo è presente una breve restituzione delle distribuzioni ottenute al variare dei parametri in input.

3 Risultati

Graficando i dati ottenuti dall'esecuzione del codice ci aspettiamo che:

1. i Suscettibili, partendo da un numero iniziale $S_i = N - I_0$, calino al crescere delle altre due categorie. Col passare dei giorni ci si aspetta che il suo valore tenda a $S_f = 0$.
2. gli Infetti, partendo da un numero pari a I_0 , crescano finché l'epidemia è in espansione (condizione riportata nel Cap.1.1) e di conseguenza, una volta raggiunto il valore massimo I_{max} , si riducano fino all'estinzione della pandemia.
3. i Rimossi, partendo da un valore $R_i = 0$, aumentino sempre di più nel corso dell'espansione della pandemia, arrivando ad un valore $R_f = N$ quando l'epidemia avrà smesso di circolare. Questo accade perché, una volta che si sono ammalate, le persone non possono più contagiarsi.

Abbiamo quindi studiato l'evoluzione di diverse tipologie di epidemia variando i parametri β e γ . In Fig.3 sono riportate le diverse distribuzioni ottenute. Per ciascun grafico, in alto, sono riportati i valori di β , γ ed il loro rapporto R_0 .

Tutti i grafici evidenziano un andamento coerente con quanto sopra riportato. Le prime due immagini mostrano valori per $\beta = 0.5, \gamma = 0.5$, in particolare, nel secondo caso, considerando solamente i primi 100 giorni per dare una visione più chiara dell'andamento. Mettiamo ora a confronto questa curva, i cui valori dei parametri sono uguali, con le curve nelle quali i parametri sono diversi e speculari.

Andando ad analizzare il grafico per $\beta = 0.7$ e $\gamma = 0.2$, si osserva un aumento degli infetti estremamente rapido, dovuto all'alto valore della probabilità di contagio. Il lento decorso

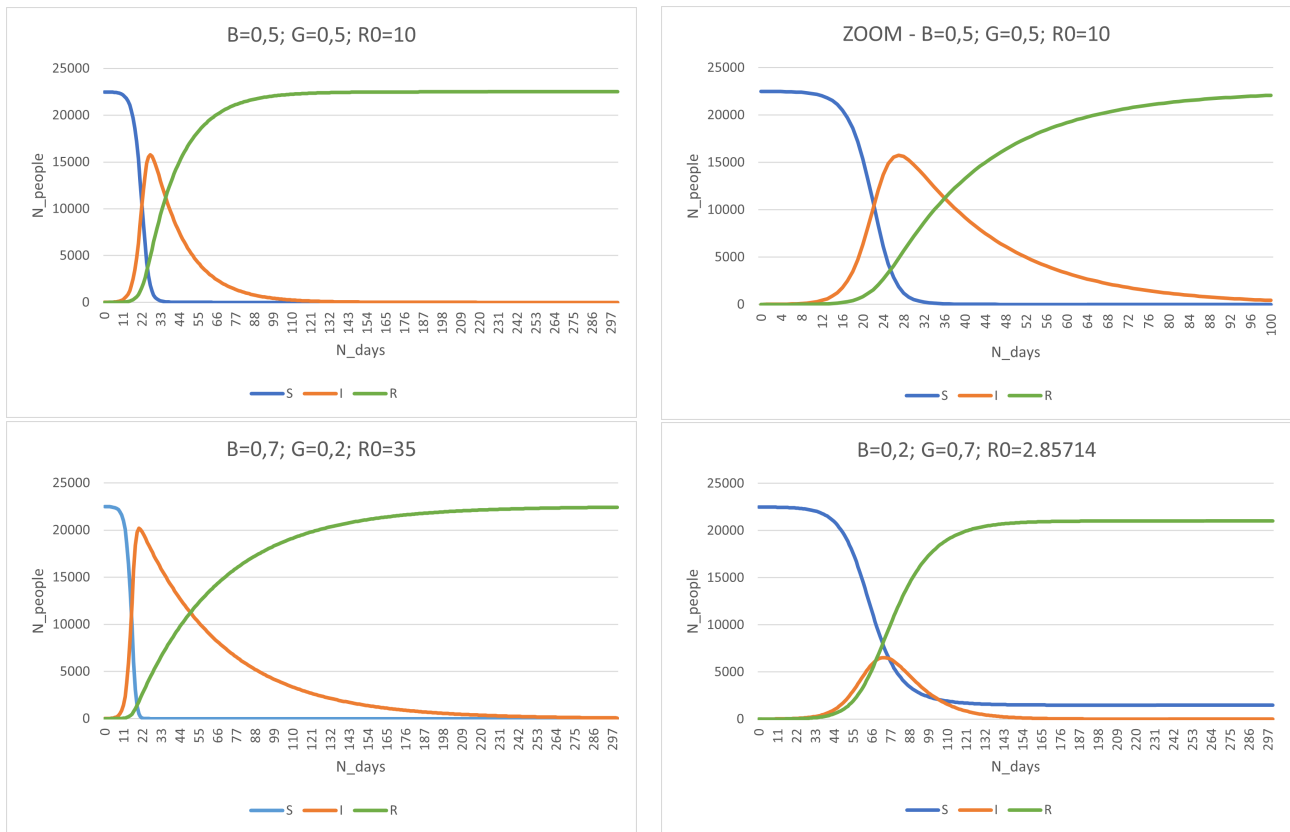


Figura 3: Rappresentazioni grafiche delle distribuzioni delle categorie S - I - R nel corso dei primi 300 giorni di pandemia (100 per l'immagine "ZOOM"). I dati sono stati presi dal file report.txt e sono stati graficati tramite Excel.

dell'epidemia è invece da attribuire al basso valore della probabilità di guarigione. Nell'ultimo grafico, con $\beta = 0.2$, $\gamma = 0.7$, abbiamo una situazione diametralmente opposta: la curva degli infetti è poco pronunciata, e decade velocemente, in accordo con quanto atteso per tali valori di β e γ .

4 Link Utili

Sotto questo *url* potrete trovare la repository in cui è contenuto il progetto

- <https://github.com/GiulaiB/SIR/tree/main>