

SuperCon2021 本選問題: 感染症流行のネットワーク 解析

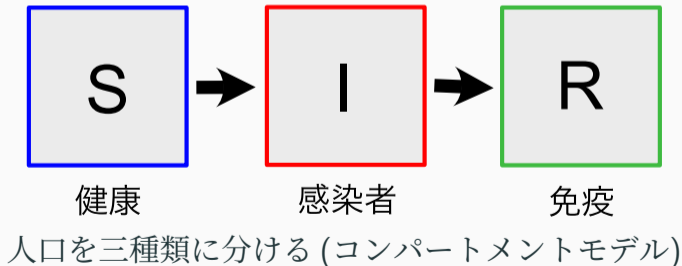
2021.8.23

はじめに: 感染症の SIR モデル

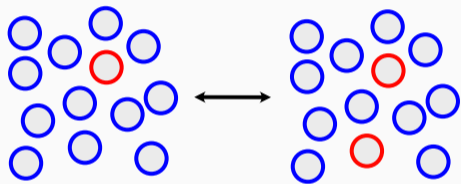
動機

- 感染症の流行を解析・予測するために数理モデルが重要な役割を果たしている。

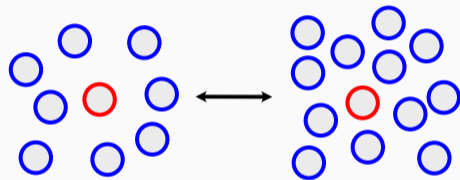
最も基本的な SIR モデル (Kermack & McKendrick, 1927)



感染者の増え方 (仮定)



感染者がひとりよりふたりの方が
二倍感染しやすい



感染者の周りに人が多い方が感染
者が増えやすい

感染者の増え方 (続き)

一日で感染する人数の考え方

- その日の感染者数 I に比例するとしよう
- その街の健康な人の数 S に比例するとしよう

一日当たり (day= t) の新たな感染者数

- $\beta S(t)I(t)$
 - β は感染率

SIR equations (simple ver.)

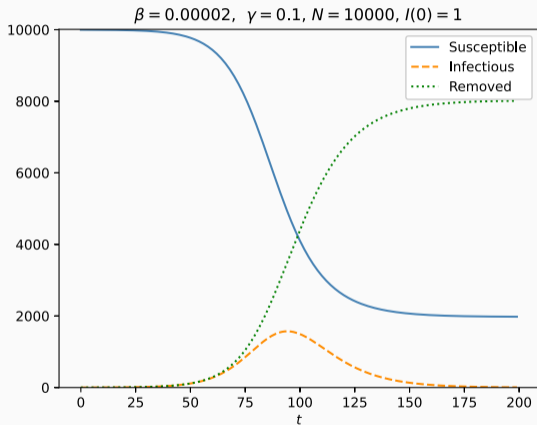
$$S(t+1) = S(t) - \beta S(t)I(t)$$

$$I(t+1) = I(t) + \beta S(t)I(t) - \gamma I(t)$$

$$R(t+1) = R(t) + \gamma I(t)$$

- 全人口 $N = S + I + R$ は一定
- γ は治癒率 (+死亡率)
- $R_0 = \frac{\beta}{\gamma} N$: 基本再生産数

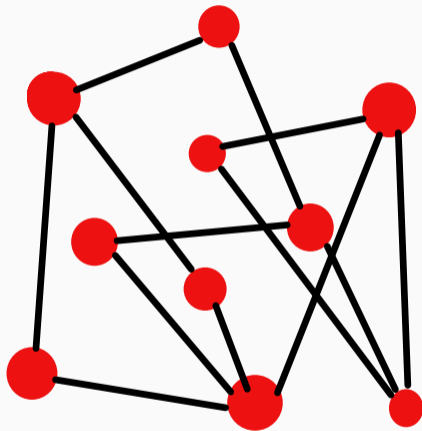
SIR モデルでの感染者数の推移



ネットワーク SIR モデル (概念)

動機

- 人々はいくつもの集団に分かれている
 - 集団内と集団間では感染症の広がりかたが違うだろう
 - つながった集団間では低い感染率 β' で感染が伝わる



10 集団 13 リンクの例 (円の大きさは集団の人数)

ネットワーク SIR モデル

モデル

$$S_i(t+1) = S_i(t) - \beta S_i(t) I_i(t) - \beta' S_i(t) \sum_{j=0}^{N_{\text{group}}-1} C_{ij} I_j(t)$$

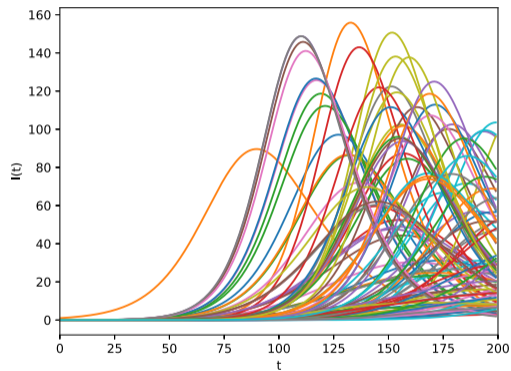
$$I_i(t+1) = I_i(t) + \beta S_i(t) I_i(t) + \beta' S_i(t) \sum_{j=0}^{N_{\text{group}}-1} C_{ij} I_j(t) - \gamma I_i(t)$$

$$R_i(t+1) = R_i(t) + \gamma I_i(t)$$

隣接行列 C_{ij}

1. 集団 i と j が直接つながっていれば $C_{ij} = 1$ かつ $C_{ij} = C_{ji}$
2. 集団 i と j が直接つながっていなければ $C_{ij} = 0$ 。

ネットワーク SIR モデルでの感染者数の推移



100 集団の例。 $t = 0$ でひとつの集団にひとりの感染者

作成するプログラム

- 各集団の人数が与えられるが、集団間のつながりは未知である。
- 第0日に0番目の集団で一人が感染者になった。その後、ネットワークSIRモデルに従って感染が広がってゆく。
- 第 $T = 200$ 日の各集団での感染者数の組 $\{I_i^{\text{prob}}(T)\} (i = 0 \sim N_{\text{group}} - 1)$ が問題として与えられるので、それを再現する隣接行列 C_{ij} を推定するプログラムを作成する

問題 (続き)

精度の定義

推定された C_{ij} を使って求められた第 T 日の各集団での感染者数の組を $\{I_i^{\text{inf}}(T)\} (i = 0 \sim N_{\text{group}} - 1)$ とすると、誤差二乗和は

$$E = \sum_{i=0}^{N_{\text{group}}-1} (I_i^{\text{inf}}(T) - I_i^{\text{prob}}(T))^2$$

で求められる。これをできるだけ小さくする。

パラメーター

- 集団の数は $N_{group} = 100$ 、日数は $T = 200$ 日に固定される。
- 感染率と回復率は $\beta = 0.0002$ 、 $\beta' = 0.000001$ 、 $\gamma = 0.1$ に固定される。
- 集団間のつながりの総数は問題ごとに異なるが、問題で与えられる (だいたい 190 程度)。

勝利条件

1. 提出されたプログラムで問題を3問解く。実行時間の上限は各問5分とする(実際には6分間走らせてタイムアウトさせ、5分以内に出力された最後の結果を採用する)。
2. 各問について、誤差二乗和が小さいものから順に順位をつけ、1位20点、2位19点、以下同様に点数化する。誤差二乗和が全く同じときには出力までの時間が短いチームを上位とする。
3. 3問の点数を合計して総合順位をつける。点数が等しい時には、3問の計算時間の合計が短いチームを上位とする。

- コンテスト専用ヘッダーファイル `sc21.h` をインクルードする。
- 入出力関数 `SC_input()` と `SC_output()` が定義されている。
- `SC_input()` はプログラム中で最初の実行文とする (MPI を使用するときは、最初の実行文が `MPI_init` にする必要があり、`SC_input()` はその直後の実行文とする)。
- `SC_output()` は何度呼んでもよいが制限時間内の最後の出力を採用する。

プログラムの提出

- プログラムのソースコードとジョブスクリプトを提出
 - ソースコードは必ずひとつのファイルにまとめてあること。分割したファイルや makefile の使用は認めない。
- 提出方法は追って discord で示す。