

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



MÔ HÌNH HÓA TOÁN HỌC

Bài tập lớn

Mô hình SIR trong dự báo COVID-19

GVHD: Nguyễn An Khương
SV thực hiện: Võ Ngọc Trọng – 1814491
Đỗ Lam Trường – 1814551
Phan Thanh Trường – 1814578
Phạm Quốc Trung – 1814522
Đặng Ngọc Tâm – 1813910

Tp. Hồ Chí Minh, Tháng 7/2020



Mục lục

1	Bài toán 1	2
1.1	Giới thiệu mô hình	2
1.2	Mô hình SIR rời rạc	2
1.3	Mô hình SIR liên tục	3
1.4	Hệ số lây nhiễm cơ bản R_o	3
2	Bài toán 2	4
2.1	Cơ sở lý thuyết của phương pháp Euler	4
2.2	Thực hiện phương pháp Euler	4
2.3	Ví dụ	6
3	Bài toán 3	7
3.1	Phương pháp suy luận Bayesian [2]	7
3.2	Thuật toán Metropolis–Hastings	8
3.3	Giải thích ý tưởng	8
3.4	Hiện thực chương trình	8
4	Bài toán 4	14
4.1	Cơ sở lý thuyết	14
4.2	Giải thích ý tưởng	14
4.3	Hiện thực chương trình	14

1 Bài toán 1

Trình bày lại chi tiết cách xây dựng mô hình SIR (cả trường hợp rời rạc lẫn liên tục) hoặc mở rộng của nó và những vấn đề liên quan.

1.1 Giới thiệu mô hình

Mô hình SIR là một mô hình toán học cơ bản về dịch bệnh, được giới thiệu trong bài báo kinh điển của Kermack và McKendrick [1]. Để xây dựng mô hình này, ta chia dân số một khu vực thành 3 nhóm sau:

- Nhóm có nguy cơ nhiễm bệnh do tiếp xúc với người bệnh (susceptible)
- Nhóm đã bị nhiễm bệnh (infected)
- Nhóm đã khỏi bệnh (recovered)

Để đơn giản, chúng ta giả sử:

- Khu vực theo dõi có dân số đông, phải được cách ly.
- Mỗi người sẽ thuộc một trong 3 nhóm người tại một thời điểm nhất định, người đã khỏi bệnh sẽ không bị lây nhiễm lại.
- Đơn vị thời gian cho mô hình là ngày (trường hợp biến rời rạc).
- Tỷ lệ tiếp xúc của mỗi người là như nhau.

Khi xây dựng mô hình, những giả sử ở trên đóng vai trò rất quan trọng. Việc chia nhóm dân cư và sự di chuyển của mỗi người trong các nhóm được tóm tắt ở hình 1



Hình 1: Mô hình SIR

1.2 Mô hình SIR rời rạc

Định nghĩa các biến dùng trong mô hình:

- **S**: số người có nguy cơ nhiễm bệnh
- **I**: số người đang nhiễm bệnh
- **R**: số người đã khỏi bệnh và chết vì bệnh

Những biến trên theo dõi số người trong các nhóm người được thể hiện trên hình 1 và đó là các biến phụ thuộc vào thời gian. Bây giờ, ta sẽ thêm hai tham số vào mô hình để mô tả cách mà một người di chuyển giữa các nhóm người có nguy cơ nhiễm bệnh, đang nhiễm bệnh và đã khỏi bệnh, kí hiệu β và γ . Hai tham số này đặc trưng cho mỗi mô hình SIR. Tham số γ thể hiện tỷ lệ hồi phục của người bị nhiễm bệnh, hay $\frac{1}{\gamma}$ là thời gian mà một người ở trong nhóm người bị bệnh. Như vậy, số người khỏi bệnh mỗi ngày được tính bằng công thức:

$$\text{số người khỏi bệnh mỗi ngày} = \gamma I \quad (1)$$

Tiếp theo, ta mô tả sự lây lan của dịch bệnh. Giả sử ta có thể tìm được số người trung bình mà mỗi người bị bệnh tiếp xúc trong suốt thời gian bị bệnh, ký hiệu c . Số người trung bình tiếp xúc với một người bị bệnh trong một ngày được tính bằng công thức: $\beta = c\gamma$. Điều này có nghĩa nếu chúng ta có I người đang nhiễm bệnh thì tổng số người tiếp xúc với những người bị nhiễm bệnh một ngày là βI . Nhưng chỉ có những người thuộc nhóm có nguy cơ lây nhiễm mới có thể bị nhiễm bệnh khi tiếp xúc với những người nhiễm bệnh, nên công thức tính số người nhiễm bệnh vào ngày mới là:

$$\text{số ca nhiễm mới mỗi ngày} = \beta \frac{S}{N} I \quad (2)$$

với N là số dân trong khu vực quan sát và tham số β là tỉ lệ tiếp xúc giữa người trong nhóm có nguy cơ nhiễm bệnh và đã nhiễm bệnh. Từ các phân tích trên, ta có được hệ phương trình biểu diễn mô hình SIR rời rạc:

$$R(t+1) = R(t) + \gamma I(t) \quad (3)$$

$$I(t+1) = I(t) - \gamma I(t) + \frac{\beta}{N} S(t) I(t) \quad (4)$$

$$S(t+1) = S(t) - \frac{\beta}{N} S(t) I(t) \quad (5)$$

$$S(t) + I(t) + R(t) = N \quad (6)$$

1.3 Mô hình SIR liên tục

Tương tự cách phân tích như mô hình SIR rời rạc, ta có hệ phương trình vi phân biểu diễn mô hình SIR liên tục:

$$\frac{dR}{dt} = \gamma I(t) \quad (7)$$

$$\frac{dI}{dt} = -\gamma I(t) + \frac{\beta}{N} S(t) I(t) \quad (8)$$

$$\frac{dS}{dt} = -\frac{\beta}{N} S(t) I(t) \quad (9)$$

$$\frac{dR}{dt} + \frac{dI}{dt} + \frac{dS}{dt} = 0 \quad (10)$$

1.4 Hệ số lây nhiễm cơ bản R_o

Một trong những đại lượng quan trọng nhất của một mô hình dự đoán dịch bệnh là hệ số lây nhiễm cơ bản R_o . Ví dụ với $R_o = 5$ thì một người bị bệnh có thể lây nhiễm cho năm người khỏe mạnh.

$$R_o = \frac{\beta}{\gamma} \quad (11)$$

Nếu $R_o > 1$, dịch bệnh sẽ lây lan nhanh và bùng phát trong cộng đồng.
Nếu $R_o < 1$, dịch bệnh sẽ khó có khả năng bùng phát.

2 Bài toán 2

Viết chương trình sử dụng thuật toán Euler tìm nghiệm của hệ SIR hoặc mở rộng của nó với tham số đầu vào gồm biến thời gian t , các hệ số tiếp xúc β , hệ số phục hồi γ và điều kiện đầu là số ca mắc bệnh $I(t_0)$ và số ca phục hồi $R(t_0)$ của mô hình tính tại thời điểm đầu tiên ghi nhận được các ca nhiễm bệnh. Giá trị trả về là mảng chứa số người nhiễm bệnh $I(t)$ và số người đã hồi phục $R(t)$ tính tại thời điểm $t \geq t_0$. Cho một số ví dụ về điều kiện đầu và các hệ số trong mô hình và dùng chương trình đã viết để tìm nghiệm xấp xỉ. Biểu diễn nghiệm xấp xỉ bằng cách vẽ đồ thị. Trường hợp là hệ SIR mở rộng như SIRD thì cần trả về $I(t)$, $R(t)$ và $D(t)$ là số ca tử vong tại thời điểm $t \geq t_0$. Trình bày chi tiết kết quả trong báo cáo

2.1 Cơ sở lý thuyết của phương pháp Euler

Phương pháp Euler là một phương pháp số bậc một để giải các phương trình vi phân thường (ODEs) với giá trị ban đầu cho trước.

Giả sử ta có phương trình vi phân bậc nhất

$$y' = f(t, y(t)) \quad (12)$$

Khi đó, áp dụng phương pháp Euler ta sẽ xấp xỉ nghiệm y bằng dãy y_n sao cho

$$y_{n+1} := y_n + f(t_n, y_n)\Delta t \quad (13)$$

với Δt là bước xấp xỉ đủ nhỏ và $f(t, y(t))$ là độ dốc của đường cong y tính tại thời điểm t . **Dạng tổng quát**

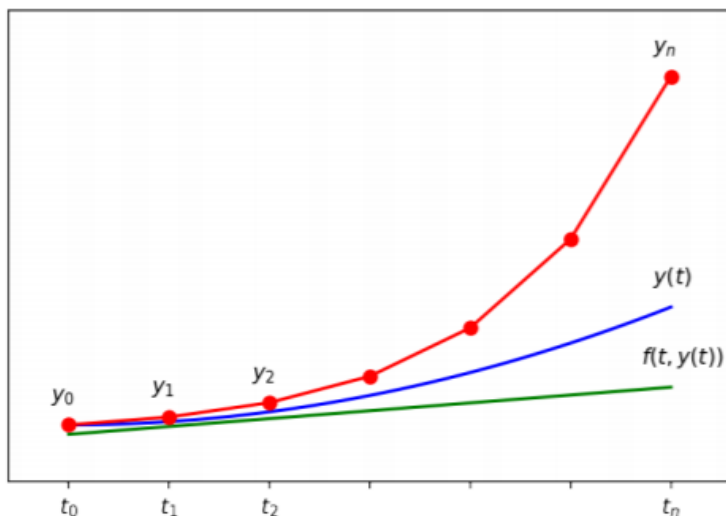
$$y'_1 = f_1(t, y_1, \dots, y_N), \quad (14)$$

$$\vdots$$

$$y'_N = f_N(t, y_1, \dots, y_N), \quad (15)$$

2.2 Thực hiện phương pháp Euler

Chương trình hiện thực phương pháp Euler trong việc giải xấp xỉ mô hình SIR.



Hình 2: Các điểm màu đỏ là các giá trị xấp xỉ đường cong y có độ dốc $f(t, y)$

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import pandas as pd
4
5 day = int(input("Nhập thời gian: "))
6 N = int(input("Nhập giá trị khởi đầu của N: "))
7 Io = int(input("Nhập giá trị khởi đầu của I: "))
8 Ro = int(input("Nhập giá trị khởi đầu của R: "))
9 dentaT = int(input("Nhập delta T: "))
10 beta = float(input("Nhập tham số beta: "))
11 gamma = float(input("Nhập tham số gamma: "))
12
13 #Euler method
14 def euler (t, beta, gamma, Io, Ro):
15     I = Io
16     R = Ro
17     S = N - I - R
18     res = []
19     rS = []
20     rI = []
21     rR = []
22     for day in range(t):
23         R_der = gamma*I
24         R_t = R_der*dentaT
25         R = R + R_t
26         I_ = I
27         I_der = beta*S*I - gamma*I
28         I_t = I_der*dentaT
29         I = I + I_t
30         S = S - beta*I_*S*dentaT
31         res.append([S, I, R])
32         rS.append(S)

```

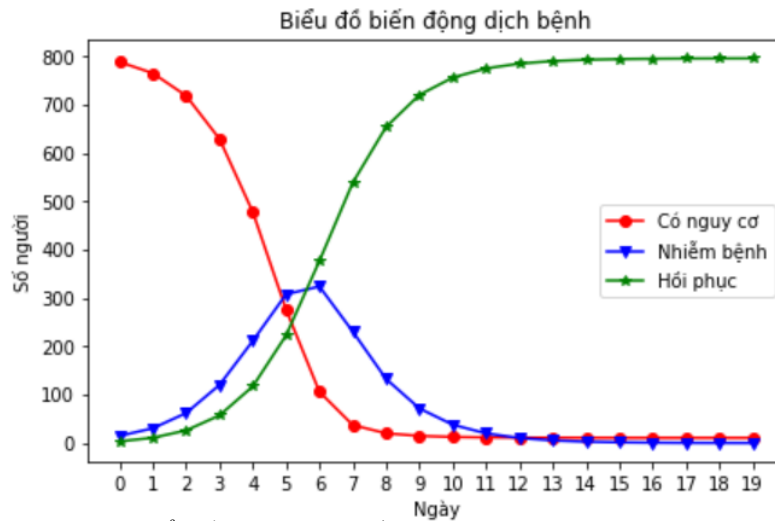
```
33     rI.append(I)
34     rR.append(R)
35     print(pd.DataFrame(res).rename(columns={0: 'Có nguy cơ', 1: 'Nhiễm bệnh', 2:
        ↪ 'Hồi phục'}))
36     return rS, rI, rR
37 listR_n, listR_t, listR_der = euler(day,beta,gamma,Io,Ro)
38 #draw plot
39 fig, axs = plt.subplots(1, 1, constrained_layout=True)
40 nums_day = np.arange(0, day, 1).tolist()
41 axs.set_title('SIR model example')
42 axs.plot(nums_day, listR_n, '-or', label="Có nguy cơ")
43 axs.plot(nums_day, listR_t, '-vb', label="Nhiễm bệnh")
44 axs.plot(nums_day, listR_der, '-*g', label="Hồi phục")
45 plt.legend(loc="center right")
46 axs.set_xlabel('Time')
47 axs.set_ylabel('Number of people')
48 plt.xticks(np.arange(0, day, 1.0))
49 plt.show()
```

2.3 Ví dụ

Kết quả chạy thử chương trình trên với bộ giá trị $(day, N, I_o, R_o, dentaT) = (20, 807, 7, 0, 1)$

	"Có nguy cơ"	"Nhiễm bệnh"	"Hồi phục"
0	788.800000	14.700000	3.500000
1	765.609280	30.540720	10.850000
2	718.844763	62.034877	26.120360
3	629.657869	120.204332	57.137799
4	478.282662	211.477373	117.239965
5	275.990740	308.030609	222.978651
6	105.963549	324.042496	376.993955
7	37.290163	230.694633	539.015203
8	20.084882	132.552598	654.362520
9	14.760276	71.600906	720.638819
10	12.646577	37.914151	756.439272
11	11.687609	19.916044	775.396347
12	11.222067	10.423564	785.354369
13	10.988119	5.445730	790.566151
14	10.868442	2.842542	793.289016
15	10.806654	1.483059	794.710287
16	10.774601	0.773583	795.451816
17	10.757931	0.403462	795.838608
18	10.749250	0.210412	796.040339
19	10.744726	0.109729	796.145544

Hình 3 thể hiện kết quả chạy của chương trình: số ca nhiễm bệnh tăng trong khoảng 6 ngày đầu và giảm dần.



Hình 3: Biểu đồ dự đoán sự biến động của dịch bệnh sử dụng mô hình SIR

3 Bài toán 3

Viết chương trình theo ngôn ngữ tự chọn để lấy mẫu sử dụng thuật toán Metropolis–Hastings với tham số đầu vào là phân bố xác suất tiên nghiệm $\pi(\beta, \gamma)$ cho trước. Giá trị trả về là một mẫu gồm các cặp β và γ có phân bố xác suất $\pi(\beta, \gamma)$. Vẽ biểu đồ thể hiện quá trình chọn mẫu. Trình bày chi tiết kết quả trong báo cáo.

3.1 Phương pháp suy luận Bayesian [2]

Để ước lượng các tham số của một mô hình dịch bệnh, ta dùng pháp suy luận Bayesian. Chúng ta cần tính giá trị $p(\theta|data)$. Áp dụng công thức xác suất có điều kiện:

$$p(\theta|data) = \frac{p(data|\theta)p(\theta)}{p(data)} \quad (16)$$

Trong đó:

- $p(\theta|data)$: xác suất hậu nghiệm của θ (posterior)
- $p(data|\theta)$: xác suất để xảy ra $data$ khi biết θ (likelihood)
- $p(\theta)$: xác suất tiên nghiệm của θ
- $p(data)$: hằng số chuẩn hóa (normalisation constant)

Cụ thể, $p(\theta)$ là độ chắc chắn của chúng ta về các giá trị của tham số dựa trên những số liệu thực nghiệm trước đó, nếu chưa có số liệu thực nghiệm, ta có thể sử dụng hàm phân phối *Beta* để biểu diễn xác suất tiên nghiệm. θ là biến ngẫu nhiên, có cùng phân phối với xác suất hậu nghiệm $p(\theta|data)$. Bằng cách quan sát số liệu thực tế, ta có thể hiệu chỉnh tham số θ của mô hình, đây cũng chính là điểm mạnh của phương pháp suy luận Bayesian. Vì $p(data)$ là hằng số, ta có thể viết lại công thức (16) như sau:

$$p(\theta|data) \propto p(data|\theta)p(\theta) \quad (17)$$

3.2 Thuật toán Metropolis–Hastings

Gọi $p(\theta)$ là hàm phân bố xác suất tiên nghiệm của θ , ý tưởng cơ bản của Metropolis–Hastings là ở mỗi bước chúng ta sẽ cân nhắc đưa ra dự định từ trạng thái hiện tại θ đến trạng thái mới θ^* với xác suất $q(\theta^*|\theta)$ khi biết trạng thái trước θ , với q được gọi là phân bố dự định (proposal distribution). Các bước của thuật toán được tiến hành như sau:

1. Khởi tạo θ_o từ phân bố xác suất tiên nghiệm $p(\theta)$, gán $\theta = \theta_o$
2. Khởi tạo ngẫu nhiên θ^* từ phân bố xác suất dự định $q(\theta^*|\theta)$
3. Nếu q là một phân bố đối xứng thỏa $q(\theta^*|\theta) = q(\theta|\theta^*)$, gán cho r xác suất giữ lại

$$r = \min(1, \frac{p(\theta^*)}{p(\theta)}) \quad (18)$$

và đi đến bước 5

4. Nếu q không là phân bố đối xứng, gán cho r xác suất giữ lại

$$r = \min(1, \frac{p(\theta^*)q(\theta|\theta^*)}{p(\theta)q(\theta^*|\theta)}) \quad (19)$$

và đi đến bước 5

5. Khởi tạo $u \sim U(0, 1)$
6. Gán

$$\theta_{(i+1)} = \begin{cases} \theta^* & u < r \\ \theta_i & u \geq r \end{cases} \quad (20)$$

$$(21)$$

7. Gán $\theta = \theta_i$

3.3 Giải thích ý tưởng

1. Đọc dữ liệu từ file, loại bỏ những giá trị ở đầu tiên vì số liệu của nó chưa thay đổi nhiều, có thể gây nhiễu.
2. Sử dụng công thức hệ SIR rời rạc để tính các cặp giá trị β và γ cho từng ngày.
3. Sử dụng từng bộ β và γ để tính tham số của từng hàm phân phối:

$$\beta \sim \Gamma(\lambda_\beta, \nu_\beta) \quad (22)$$

$$\gamma \sim \Gamma(\lambda_\gamma, \nu_\gamma) \quad (23)$$

4. Hiện thực giải thuật Metropolis–Hastings sử dụng các cặp tham số vừa tìm được: $(\lambda_\beta, \nu_\beta)$, $(\lambda_\gamma, \nu_\gamma)$.

3.4 Hiện thực chương trình

Sử dụng dữ liệu nước Mỹ để thực hiện tính toán.

```
1 import numpy as np
2 import math
3
4 import pandas as pd
5 from scipy.stats import gamma
6
7 import matplotlib.pyplot as plt
8 import scipy.special as sps
9 import seaborn as sns
10
11 # read data and assign to list S, I, R #####
12 dataconfirm = pd.read_csv("time_series_covid19_confirmed_global.csv")
13 dataconfirm.head()
14 confirm = dataconfirm.loc[dataconfirm['Country/Region'] == 'US']
15
16 dataDead = pd.read_csv("time_series_covid19_deaths_global.csv")
17 dataDead.head()
18 dead = dataDead.loc[dataDead['Country/Region'] == 'US']
19
20 dataRecover = pd.read_csv("time_series_covid19_recovered_global.csv")
21 dataRecover.head()
22 recover = dataRecover.loc[dataRecover['Country/Region'] == 'US']
23
24 cfUS = confirm.values.tolist()[0]
25 for i in range(84):
26     cfUS.pop(0)
27
28 deadUS = dead.values.tolist()[0]
29 for i in range(84):
30     deadUS.pop(0)
31
32 recoverUS = recover.values.tolist()[0]
33 for i in range(84):
34     recoverUS.pop(0)
35
36 N = 328234567 ## population of region
37 I = cfUS
38 R = deadUS
39 for i in range(len(I)):
40     R[i] = R[i] + recoverUS[i]
41 S = [N for i in range(len(I))]
42 for i in range(len(I)):
43     S[i] = N - I[i] - R[i]
44 #####
45 # Estimate gamma and beta from data
46 gamma_ = []
47 beta_ = []
48 for i in range(len(I)-1):
49     gamma_.append((R[i+1]-R[i])/I[i])
50     beta_.append(N*(I[i+1]- I[i]+R[i+1] - R[i])/I[i]/S[i])
51 #####
52 # Calculate lama and muy from list of gamma, beta
```

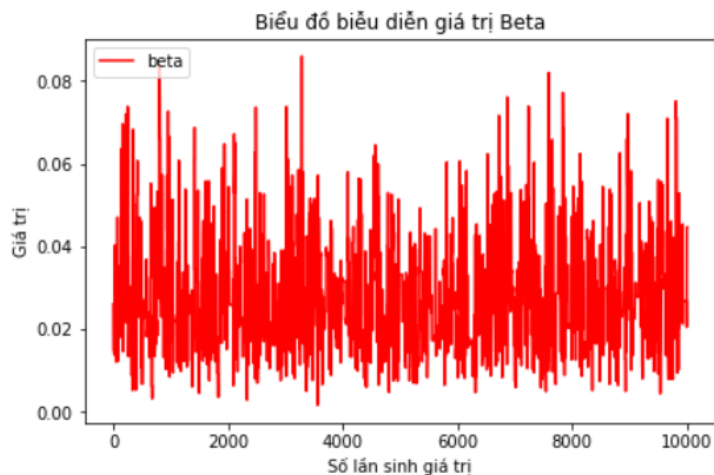
```
53 def MLEgamma(x):
54     x = pd.Series(x)
55     mean = x.mean()
56     var = x.var()
57     lamda = (mean**2)/var
58     muy = lamda / mean
59     return lamda , muy
60
61 lamdaBeta, muyBeta = MLEgamma(beta_)
62 lamdaGamma, muyGamma = MLEgamma(gamma_)
63
64 #####
65 # Calculate pi(beta,gamma) from lamda and muy
66
67 def calculateProbability(B,G,lamda_B, muy_B, lamda_G, muy_G):
68     pBeta = pow(muy_B,lamda_B)/math.gamma(lamda_B) * pow(B,lamda_B - 1)*
69     ↪ math.exp(-B*muy_B)
70     pGamma = pow(muy_G,lamda_G)/math.gamma(lamda_G) * pow(G,lamda_G - 1)*
71     ↪ math.exp(-G*muy_G)
72     return pBeta*pGamma
73
74 #####3
75 # metropolis_hasting function
76
77 def metropolis_hasting(lamda_B, muy_B, lamda_G, muy_G, size):
78     [Bo,Go] = [np.random.gamma(lamda_B ,1/muy_B) , np.random.gamma(lamda_G ,1/muy_G)]
79     listB = []
80     listG = []
81     [B,G] = [Bo,Go]
82     p = calculateProbability(B,G,lamda_B,muy_B,lamda_G,muy_G)
83     for i in range(size):
84         B1,G1 = (B + np.random.normal(0, 0.03)) ,(G + np.random.normal(0, 0.02))
85         listB.append(B)
86         listG.append(G)
87         if (B1 < 0 or G1 < 0):
88             continue
89         p1 = calculateProbability(B1,G1,lamda_B,muy_B,lamda_G,muy_G)
90         r = min(1,p1/p)
91         q = np.random.uniform(0,1)
92         if (q < r):
93             [B,G] = [B1,G1]
94             p = p1
95         return listB, listG
96
97 #####
98 # Draw plot
99
100 lB, lG = metropolis_hasting(lamdaBeta,muyBeta, lamdaGamma, muyGamma, 10000)
101
102 fig, axs = plt.subplots(1, 1, constrained_layout=True)
103 axs.set_title('Biểu đồ biểu diễn giá trị Beta và Gamma')
104 axs.plot(lB, 'r', label = "beta")
105 axs.plot(lG, 'b', label = "gamma")
```

```

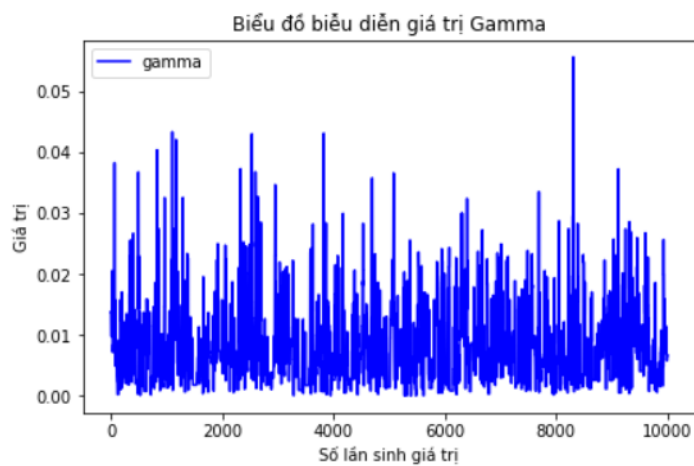
103  axs.set_xlabel('Số lần sinh giá trị')
104  axs.set_ylabel('Giá trị')
105  axs.legend(loc = "upper left")
106  plt.show()
107
108  #####
109  # Draw plot 2
110  data = []
111  for i in range(len(lB)):
112      data.append([lB[i],lG[i]])
113  df = pd.DataFrame(data, columns=["beta", "gamma"])
114
115  g = sns.jointplot(x="beta", y="gamma", data=df, kind="kde", color="r")
116  g.plot_joint(plt.scatter, c="b", s=30, linewidth=1, marker="+")
117  g.ax_joint.collections[0].set_alpha(0)
118  g.set_axis_labels("$beta$", "$gamma$");

```

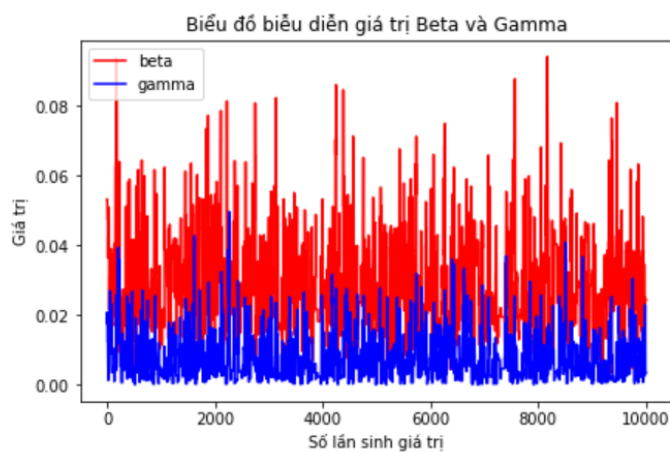
Hình 4 và hình 5 lần lượt thể hiện giá trị của β và γ sau khi random. Hình 6 và 7 thể hiện giá trị β và γ sau khi tạo random theo phương pháp.



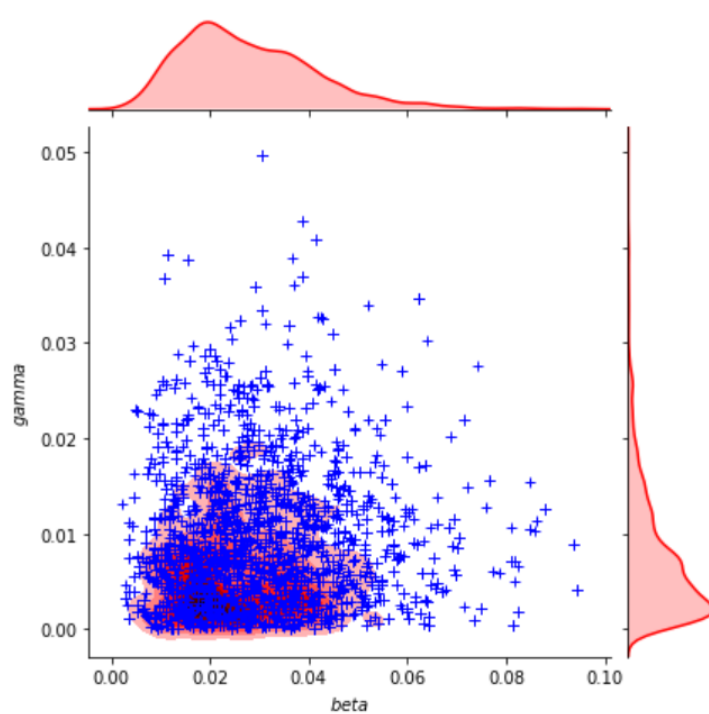
Hình 4: Biểu đồ thể hiện giá trị β sau khi random



Hình 5: Biểu đồ thể hiện giá trị γ sau khi tạo random



Hình 6: Biểu đồ thể hiện giá trị của β và γ sau khi random



Hình 7: Biểu đồ thể hiện giá trị của β và γ sau khi random

4 Bài toán 4

4.1 Cơ sở lý thuyết

Tiếp tục phần suy luận Bayesian ở phần 3.1, để tính giá trị kì vọng của đặc trưng của một mô hình, ta lấy tích phân:

$$E[A] = \int p(\theta|data)X(\theta)d\theta \quad (24)$$

Ví dụ với mô hình SIR có hai tham số β và γ , giá trị kì vọng của hệ số lây nhiễm cơ bản R_o được tính như sau:

$$E[R_o] = \int p(\beta, \gamma|data) \frac{\beta}{\gamma} d\beta d\gamma \quad (25)$$

Thông thường, việc xác định một biểu thức cho $p(\beta, \gamma|data)$ là không thể, nên ta sẽ dùng xấp xỉ Monte-Carlo:

$$E[R_o] = \int p(\beta, \gamma|data) \frac{\beta}{\gamma} d\beta d\gamma \propto \int p(data|\beta, \gamma) \frac{\beta}{\gamma} d\beta d\gamma \approx \sum_{i=1}^m p(data|\beta, \gamma) \frac{\beta_i}{\gamma_i} \quad (26)$$

4.2 Giải thích ý tưởng

Để tính được chính xác giá trị trung bình của R_o , ta cần biết được giá trị của $\pi(data)$, nhưng mà trong thực tế, giá trị này rất khó để tính được. Khi hiện thực, ta chỉ có thể tính được một giá trị tỉ lệ với R_o .

$$E(R_o) = \int \pi(\beta, \gamma|data) R_o(\beta, \gamma) d(\beta, \gamma) \propto \int \pi(data|\beta, \gamma) \pi(\beta, \gamma) R_o(\beta, \gamma) d(\beta, \gamma) \quad (27)$$

Giá trị này có thể được xấp xỉ bằng:

$$\sum_{i=1}^m \pi(data|\beta, \gamma) \frac{\beta_i}{\gamma_i}$$

Trong đó, $\pi(data|\beta, \gamma)$ được tính bằng:

$$\pi(data|\beta, \gamma) = \prod_{i=1}^n f(data(t_i)|\beta, \gamma) = \prod_{i=1}^n \frac{\gamma^\beta}{\Gamma(\beta)} data(t_i)^{\beta-1} \exp\{-\gamma data(t_i)\} \quad (28)$$

Tuy nhiên, giá trị $\pi(data|\beta, \gamma)$ rất nhỏ nên kết quả nhận được sau tính toán sẽ gần bằng 0. Ta không thể so sánh được giữa các nước với nhau vì kết quả đã được làm tròn về 0 bởi ngôn ngữ lập trình.

Do đó, để có thể so sánh được mức độ bùng phát dịch và cách mà các nước thực hiện chế độ cách ly, chữa trị, nhóm em đề xuất một giá trị r để so sánh được tính toán như sau:

- Tính tổng $\frac{\beta}{\gamma}$ từ mẫu được sinh ra ở câu 3.
- Chia tổng này cho số lượng β (hoặc γ)

4.3 Hiện thực chương trình

```
1 list_b , list_g = metropolis_hasting(lamdaBeta,muyBeta, lamdaGamma, muyGamma,  
  ↳ 10000)  
2 r = 0.0  
3 for i in range(len(list_b)):  
4     r = r + list_b[i]/list_g[i]  
5 print ( r/len(list_b) )
```

Kết quả chạy thử:

- Với dữ liệu của Mỹ, giá trị $r = 15.914473523842096$
- Với dữ liệu của Singapore, giá trị $r = 7.06919973929339$

Ở Mỹ, do văn hóa đề cao tính tự do cá nhân nên thời gian đầu mọi người không đeo khẩu trang dẫn đến dịch bùng phát nhanh và kéo dài. Ở Singapore, chính phủ có chính sách chống dịch tốt hiệu quả hơn nên có thể kiểm soát được dịch bệnh, tuy nhiên, sau tháng 4, do sự chủ quan của người dân nên dịch bùng phát trở lại. Dẫu vậy, giá trị r của Singapore vẫn thấp hơn giá trị r của Mỹ, chứng tỏ dịch bệnh ở Mỹ bùng phát rất mạnh và gây ảnh hưởng nghiêm trọng đến sức khỏe và cuộc sống của người dân.



Tài liệu

- [1] William Ogilvy Kermack and Anderson G McKendrick. “A contribution to the mathematical theory of epidemics”. In: *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* 115.772 (1927), pp. 700–721.
- [2] Anton Camacho Sebastian Funk Gwen Knight. *Model fitting and inference for infectious disease dynamics*. URL: https://www.imsc.res.in/~sitabhra/meetings/infection15/talks/intro_inference_chennai_slides.pdf.