Homework 4: Forecasting, Interventions, Immunization

Released: Oct 31, 2024; Due: **5pm ET**, Nov 14, 2024

Georgia Tech
College of Computing
B. Aditya Prakash

CSE 8803 EPI Fall 2024
Student NAME: Shikhar Verma
Student GTID: 903497421

Reminders:

1. *Out of 100 points. 4 Questions. Contains 5 pages.*
2. *If you use Late days, mark how many you are using (out of maximum 4 available) at the top of your answer PDF.*
3. *There could be more than one correct answer. We shall accept them all.*
4. *Whenever you are making an assumption, please state it clearly.*
5. *You will submit a solution pdf* `LASTNAME.pdf` *containing your answers and the plots as well as a tar-ball* `LASTNAME.tgz` *that contains your code and any output files.*
6. *Please type your answers either in LaTeXdocument or in a separate file like a Word document and then convert it into a pdf file. Typed answers are strongly encouraged. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.*
7. *Additionally, you will submit one tar-ball* `LASTNAME.tgz` *that contains your code and any results files. Code and results for each question should be contained in a separate sub-directory (Eg:* `Q1`*) and there should be a* `README.txt` *file for each sub-directory explaining any packages to install, command to run the code files and location of the expected output. Please follow the naming convention* ***strictly****.*
8. *If a question asks you to submit code please enter the file path (Eg:* `Q1/Q-1.3.1.py`*) in the solution pdf.*
9. *You can download all the datasets needed for this homework from canvas files, you can check the information about the datasets in the* `README.txt` *file.*

1. (12 points) Immunization in network models

Consider the so-called acquaintance immunization policy (where we pick a uniformly random neighbor of a uniformly random node) and the uniformly random strategy (of picking a node at random). For each policy, you should keep sampling till you have picked k different nodes (where k is the budget). Let us call the former the FRIENDS policy and the latter the RANDOM policy. We want to understand the relative performance of these two policies.

Generate an undirected unweighted graph: G_{ba}. G_{ba} is a Barabasi-Albert preferential model graph with $n = 1000$ nodes (steps) and the number of edges to attach at each step $m = 2$ (You can set up as `G_ba = nx.barabasi_albert_graph(1000,2)`). In the Barabasi-Albert preferential model the probability that a new vertex attaches to a given old vertex is proportional to the (total) vertex degree.

Q 1.1 (4 points) Use can use the implementation from `sis_model.py` provided in canvas. Set $\beta = 0.2, \delta = 0.2$, and `max_time`=500. Initialize the model with all nodes as infected at time-step 0. Run it 200 times on G_{ba}. Report the average number of infected nodes at each step till max_time in the report PDF.

Solution:
Time Step: Average Infected

0: 1000.0, 1: 897.0, 2: 822.0, 3: 777.0, 4: 741.0, 5: 738.0, 6: 732.0, 7: 731.0, 8: 723.0, 9: 719.0, 10: 714.0, 11: 699.0, 12: 698.0, 13: 683.0, 14: 691.0, 15: 698.0, 16:

691.0, 17: 686.0, 18: 678.0, 19: 674.0, 20: 690.0, 21: 669.0, 22: 661.0, 23: 667.0, 24: 686.0, 25: 674.0, 26: 681.0, 27: 660.0, 28: 664.0, 29: 687.0, 30: 671.0, 31: 663.0, 32: 671.0, 33: 687.0, 34: 684.0, 35: 701.0, 36: 691.0, 37: 682.0, 38: 688.0, 39: 689.0, 40: 688.0, 41: 685.0, 42: 687.0, 43: 679.0, 44: 679.0, 45: 671.0, 46: 681.0, 47: 690.0, 48: 695.0, 49: 679.0, 50: 686.0, 51: 688.0, 52: 683.0, 53: 694.0, 54: 701.0, 55: 698.0, 56: 704.0, 57: 706.0, 58: 714.0, 59: 689.0, 60: 698.0, 61: 690.0, 62: 674.0, 63: 676.0, 64: 672.0, 65: 697.0, 66: 688.0, 67: 693.0, 68: 702.0, 69: 694.0, 70: 704.0, 71: 702.0, 72: 703.0, 73: 711.0, 74: 705.0, 75: 705.0, 76: 690.0, 77: 701.0, 78: 690.0, 79: 674.0, 80: 662.0, 81: 667.0, 82: 673.0, 83: 672.0, 84: 654.0, 85: 655.0, 86: 668.0, 87: 679.0, 88: 684.0, 89: 678.0, 90: 688.0, 91: 707.0, 92: 715.0, 93: 692.0, 94: 681.0, 95: 690.0, 96: 689.0, 97: 683.0, 98: 680.0, 99: 674.0, 100: 666.0, 101: 671.0, 102: 684.0, 103: 680.0, 104: 695.0, 105: 689.0, 106: 676.0, 107: 669.0, 108: 680.0, 109: 700.0, 110: 679.0, 111: 682.0, 112: 680.0, 113: 697.0, 114: 698.0, 115: 688.0, 116: 691.0, 117: 677.0, 118: 698.0, 119: 698.0, 120: 702.0, 121: 693.0, 122: 682.0, 123: 666.0, 124: 664.0, 125: 654.0, 126: 658.0, 127: 669.0, 128: 668.0, 129: 665.0, 130: 667.0, 131: 661.0, 132: 669.0, 133: 685.0, 134: 697.0, 135: 696.0, 136: 690.0, 137: 683.0, 138: 699.0, 139: 685.0, 140: 691.0, 141: 676.0, 142: 682.0, 143: 679.0, 144: 678.0, 145: 696.0, 146: 691.0, 147: 678.0, 148: 659.0, 149: 665.0, 150: 679.0, 151: 673.0, 152: 670.0, 153: 679.0, 154: 669.0, 155: 685.0, 156: 689.0, 157: 687.0, 158: 686.0, 159: 668.0, 160: 670.0, 161: 679.0, 162: 675.0, 163: 663.0, 164: 678.0, 165: 680.0, 166: 683.0, 167: 698.0, 168: 676.0, 169: 674.0, 170: 697.0, 171: 681.0, 172: 666.0, 173: 669.0, 174: 678.0, 175: 657.0, 176: 658.0, 177: 681.0, 178: 691.0, 179: 686.0, 180: 678.0, 181: 667.0, 182: 678.0, 183: 679.0, 184: 687.0, 185: 690.0, 186: 717.0, 187: 702.0, 188: 706.0, 189: 710.0, 190: 710.0, 191: 688.0, 192: 673.0, 193: 679.0, 194: 692.0, 195: 688.0, 196: 681.0, 197: 708.0, 198: 694.0, 199: 691.0, 200: 682.0, 201: 682.0, 202: 679.0, 203: 673.0, 204: 675.0, 205: 690.0, 206: 679.0, 207: 698.0, 208: 688.0, 209: 665.0, 210: 657.0, 211: 661.0, 212: 669.0, 213: 670.0, 214: 660.0, 215: 673.0, 216: 667.0, 217: 675.0, 218: 670.0, 219: 669.0, 220: 679.0, 221: 696.0, 222: 703.0, 223: 693.0, 224: 672.0, 225: 680.0, 226: 674.0, 227: 674.0, 228: 682.0, 229: 663.0, 230: 666.0, 231: 663.0, 232: 674.0, 233: 663.0, 234: 660.0, 235: 659.0, 236: 655.0, 237: 666.0, 238: 670.0, 239: 664.0, 240: 676.0, 241: 669.0, 242: 675.0, 243: 685.0, 244: 687.0, 245: 703.0, 246: 680.0, 247: 700.0, 248: 693.0, 249: 680.0, 250: 681.0, 251: 691.0, 252: 710.0, 253: 704.0, 254: 697.0, 255: 697.0, 256: 689.0, 257: 708.0, 258: 694.0, 259: 687.0, 260: 696.0, 261: 683.0, 262: 669.0, 263: 674.0, 264: 677.0, 265: 672.0, 266: 682.0, 267: 685.0, 268: 702.0, 269: 706.0, 270: 701.0, 271: 707.0, 272: 696.0, 273: 688.0, 274: 706.0, 275: 713.0, 276: 686.0, 277: 693.0, 278: 685.0, 279: 686.0, 280: 711.0, 281: 687.0, 282: 699.0, 283: 700.0, 284: 681.0, 285: 688.0, 286: 657.0, 287: 660.0, 288: 660.0, 289: 654.0, 290: 662.0, 291: 664.0, 292: 679.0, 293: 668.0, 294: 655.0, 295: 658.0, 296: 680.0, 297: 678.0, 298: 699.0, 299: 709.0, 300: 703.0, 301: 698.0, 302: 685.0, 303: 694.0, 304: 676.0, 305: 677.0, 306: 684.0, 307: 687.0, 308: 668.0, 309: 681.0, 310: 686.0, 311: 670.0, 312: 678.0, 313: 676.0, 314: 675.0, 315: 694.0, 316: 691.0, 317: 688.0, 318: 678.0, 319: 678.0, 320: 687.0, 321: 676.0, 322: 677.0, 323: 673.0, 324: 679.0, 325: 679.0, 326: 676.0, 327: 672.0, 328: 690.0, 329: 685.0, 330: 692.0, 331: 679.0, 332: 678.0, 333: 698.0, 334: 694.0, 335: 689.0, 336: 686.0, 337: 693.0, 338: 684.0, 339: 697.0, 340: 690.0, 341: 688.0, 342: 693.0, 343: 704.0, 344: 713.0, 345: 707.0, 346: 717.0, 347: 719.0, 348: 715.0, 349: 708.0, 350: 685.0, 351: 680.0, 352: 665.0, 353: 656.0, 354: 662.0, 355: 666.0, 356: 679.0, 357: 679.0,

358: 688.0, 359: 696.0, 360: 705.0, 361: 697.0, 362: 698.0, 363: 693.0, 364: 675.0, 365: 672.0, 366: 681.0, 367: 700.0, 368: 694.0, 369: 698.0, 370: 681.0, 371: 687.0, 372: 694.0, 373: 695.0, 374: 674.0, 375: 675.0, 376: 681.0, 377: 695.0, 378: 691.0, 379: 673.0, 380: 681.0, 381: 670.0, 382: 672.0, 383: 688.0, 384: 702.0, 385: 695.0, 386: 693.0, 387: 689.0, 388: 680.0, 389: 683.0, 390: 664.0, 391: 662.0, 392: 675.0, 393: 680.0, 394: 699.0, 395: 668.0, 396: 659.0, 397: 668.0, 398: 653.0, 399: 669.0, 400: 683.0, 401: 685.0, 402: 679.0, 403: 662.0, 404: 687.0, 405: 691.0, 406: 704.0, 407: 708.0, 408: 696.0, 409: 704.0, 410: 704.0, 411: 723.0, 412: 713.0, 413: 717.0, 414: 697.0, 415: 705.0, 416: 696.0, 417: 687.0, 418: 677.0, 419: 679.0, 420: 674.0, 421: 665.0, 422: 669.0, 423: 677.0, 424: 686.0, 425: 689.0, 426: 694.0, 427: 677.0, 428: 697.0, 429: 684.0, 430: 698.0, 431: 691.0, 432: 681.0, 433: 677.0, 434: 692.0, 435: 693.0, 436: 689.0, 437: 692.0, 438: 717.0, 439: 688.0, 440: 686.0, 441: 683.0, 442: 688.0, 443: 690.0, 444: 679.0, 445: 696.0, 446: 666.0, 447: 690.0, 448: 682.0, 449: 670.0, 450: 692.0, 451: 687.0, 452: 680.0, 453: 679.0, 454: 686.0, 455: 689.0, 456: 689.0, 457: 693.0, 458: 674.0, 459: 684.0, 460: 681.0, 461: 661.0, 462: 675.0, 463: 665.0, 464: 666.0, 465: 657.0, 466: 667.0, 467: 669.0, 468: 664.0, 469: 665.0, 470: 679.0, 471: 658.0, 472: 661.0, 473: 656.0, 474: 655.0, 475: 656.0, 476: 661.0, 477: 669.0, 478: 679.0, 479: 669.0, 480: 662.0, 481: 673.0, 482: 673.0, 483: 702.0, 484: 701.0, 485: 707.0, 486: 696.0, 487: 691.0, 488: 672.0, 489: 661.0, 490: 677.0, 491: 682.0, 492: 692.0, 493: 696.0, 494: 683.0, 495: 691.0, 496: 691.0, 497: 678.0, 498: 685.0, 499: 703.0, 500: 700.0

Q 1.2 (2 points) Use your implementation of FRIENDS and RANDOM in HW3 or sampling functions provided in `util.py`. Given the budget, $k = 100$, report the nodes chosen according to each policy in the report PDF.

Solution:
RANDOM POLICY: [521, 737, 740, 660, 411, 678, 626, 513, 859, 136, 811, 76, 636, 973, 938, 899, 280, 883, 761, 319, 549, 174, 371, 527, 210, 235, 101, 986, 902, 947, 346, 139, 621, 499, 370, 198, 687, 584, 901, 59, 328, 96, 312, 974, 299, 277, 924, 601, 439, 837, 570, 879, 261, 578, 23, 30, 617, 10, 221, 820, 296, 54, 542, 209, 604, 692, 662, 866, 70, 543, 107, 493, 590, 741, 292, 289, 652, 39, 589, 307, 679, 66, 275, 67, 318, 548, 998, 714, 753, 327, 382, 451, 522, 218, 787, 436, 764, 88, 63, 826]

NEIGHBOR POLICY: [910, 307, 341, 48, 360, 709, 146, 147, 157, 174, 4, 29, 238, 176, 93, 49, 497, 424, 824, 468, 87, 11, 209, 4, 200, 356, 4, 446, 107, 243, 48, 11, 4, 190, 704, 216, 0, 92, 827, 978, 348, 228, 102, 55, 3, 210, 47, 328, 177, 45, 35, 171, 580, 872, 6, 844, 328, 39, 528, 42, 721, 28, 61, 411, 662, 811, 710, 256, 790, 9, 39, 145, 907, 65, 307, 661, 880, 4, 91, 935, 36, 258, 168, 777, 363, 6, 332, 214, 474, 2, 600, 438, 37, 475, 629, 161, 591, 809, 7, 64]

Q 1.3 (4 points) Run the SIS model with $\beta = \delta = 0.2$ on G_{ba} from Q1.1. Pick $k = 100$ nodes according to both FRIENDS and RANDOM policies (use the nodes from Q1.2). Remove these nodes from the G_{ba}, and re-run the SIS model on the new (smaller) versions of each graph. Generate plots: plot the average number of infections vs time when (a) no nodes have been removed (b) when nodes have been removed according to FRIENDS and (c) when nodes have been removed according to RANDOM (use different colors for each line (a)-(b)-(c)). Attach the plots in the reported PDF. Note:

You should run 50 times and take the average for each line a-b-c)

Solution:

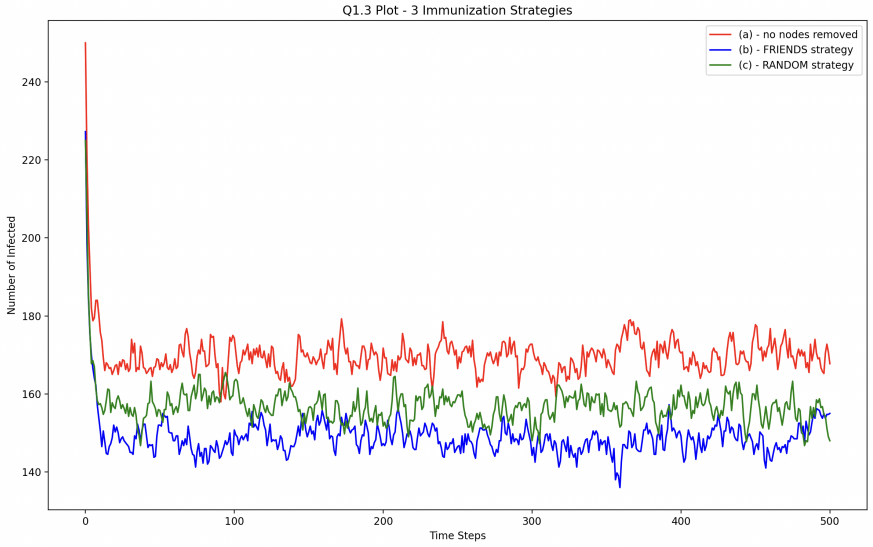

Figure 1: Q1.3

Q 1.4 (2 points) What do you observe w.r.t. the performance of RANDOM and FRIENDS? Explain your observations in the report PDF.

Solution:
It seems that the number of infections is the highest for line (a) while the number of infections is the lowest for line (b). In other words, the FRIENDS strategy seemed to work better than the RANDOM strategy, and both vaccination strategies had a lower number of infections in comparison to no vaccination at all. The FRIENDS strategy performs better than the RANDOM strategy because it uses relationship-based data to make its predictions which are better aligned with real-world social interactions.

2. (40 points) Vaccination interventions in ODE model using Multi-arm bandits

In this question, we look at a simplified toy example to study formulating vaccination policy planning as a multi-arm bandit problem. We will first set up a variant of the SIR model that accounts for the influence of vaccination rate on disease spread. Next, we will set the problem of choosing an optimal vaccination strategy for our SIR model. Finally, we will use various multi-arm bandit strategies to solve for optimal vaccination strategy.

We will simulate a variation of the SIR model to study the effect of different levels of vaccination intervention. Let $S(t), I(t)$ and $R(t)$ be the fraction of the population in susceptible, infected, and recovered state. We will divide the population into vaccinated and unvaccinated.

Let $S_1(t), I_1(t)$ and $R_1(t)$ be a fraction of the total population that is susceptible, infected, and recovered as well as unvaccinated. Similarly, let $S_2(t), I_2(t)$ and $R_2(t)$ be a fraction of

the population that is susceptible, infected, and recovered as well as vaccinated. Therefore, we have $S_1(t) + S_2(t) = S(t)$ and similarly for infected and recovered states.

We define the SIR model via the following ODE equations:

$$\begin{aligned}
\frac{dS_1}{dt} &= -\beta S_1(I_1 + I_2) \\
\frac{dI_1}{dt} &= \beta S_1(I_1 + I_2) - \gamma I_1 \\
\frac{dR_1}{dt} &= \gamma I_1 \\
\frac{dS_2}{dt} &= -\beta(1-\rho) S_2(I_1 + I_2) \\
\frac{dI_2}{dt} &= \beta(1-\rho) S_2(I_1 + I_2) - \gamma(1-\rho) I_2 \\
\frac{dR_2}{dt} &= \gamma(1-\rho) I_2
\end{aligned} \tag{1}$$

where β and γ are the usual SIR model parameters and ρ determines the effectiveness of the vaccine in both reducing the rate of infection and probability of transitioning to R state among the vaccinated.

We have provided a boilerplate code in `datasets/q2.ipynb` notebook and you only need to fill in the requested portions.

Q 2.1 (6 points) Implement the above-defined SIR model and submit the code. Specifically, complete the `model_ode` function in the notebook.

Let 90% of the population be susceptible initially and the rest 5% be infected. Set $\beta = 0.1$, $\gamma = 0.01$ and $\rho = 0.3$. Let $k = 50\%$ of both infected and susceptible populations be vaccinated (i.e., $S_1(0) = S_2(0) = 0.45$ and $I_1(0) = I_2(0) = 0.05$. Set $T = 200$ and plot the fraction of each compartment for time-steps from 0 to T. Now set $k = 10\%$ repeat the ODE simulation and plot the fraction of each compartment for time-steps from 0 to T. How does the fraction of the population $R(T) = R_1(T) + R_2(T)$ at the end ($T = 200$) change with k?

Hint: Refer to Q1 in HW 1.

Solution:

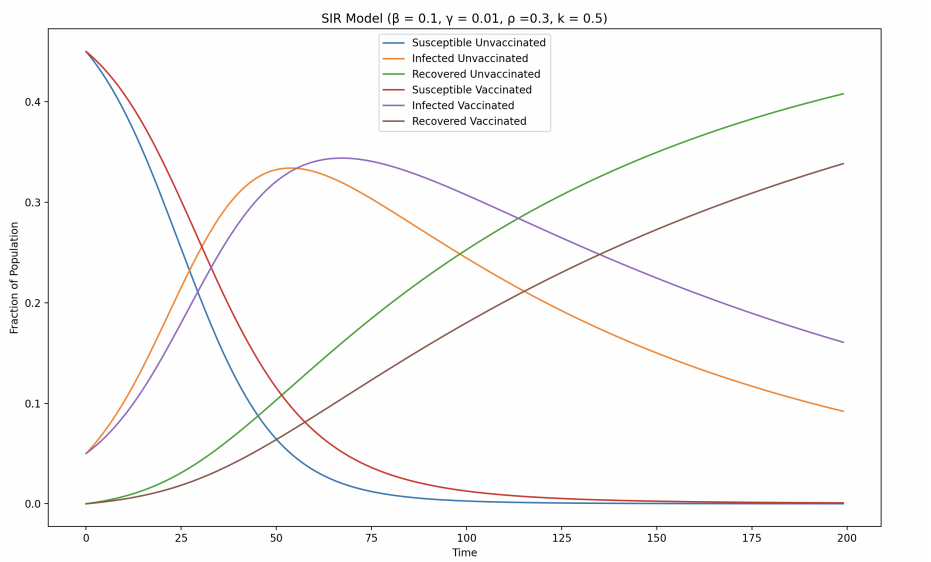

Figure 2: Q2.1 Plot 1

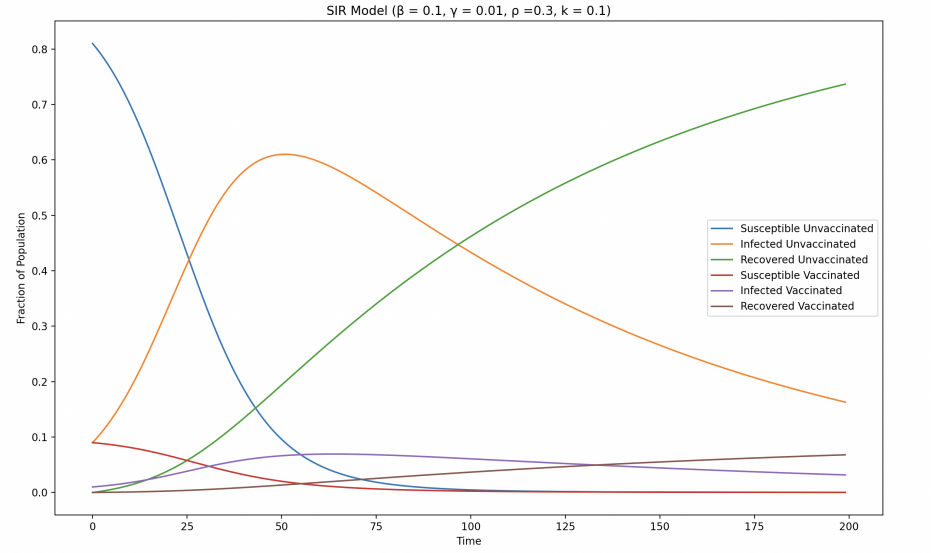

Figure 3: Q2.1 Plot 2

When $k = 0.5$, tthe $R(200)$ value was around 0.75 while when $k = 0.1$, the $R(200)$ value was around 0.81. It seems as that k decreases, the fraction of people in the recovered state for both populations increased.

Q 2.2 (6 points) In many cases, we are not certain about the parameters β, γ, ρ of the model. We model them as random variables. Assume that $\beta \sim Uniform(0.05, 0.15)$, $\gamma \sim Uniform(0.005, 0.015)$ and $\epsilon \sim Uniform(0.1, 0.3)$. Complete the `stochastic_model_oracle` function.

For each of $k = [0\%, 10\%, 20\%, \ldots, 90\%, 100\%]$ run the SIR model for 1000 runs, sam-

pling the parameters at beginning of each run. Compute the average of $R(T)$ for each of the values of k. Submit a plot with x-axis as k and y-axis as mean $R(T)$ (averaged over 1000 runs for each values of k).

Solution:

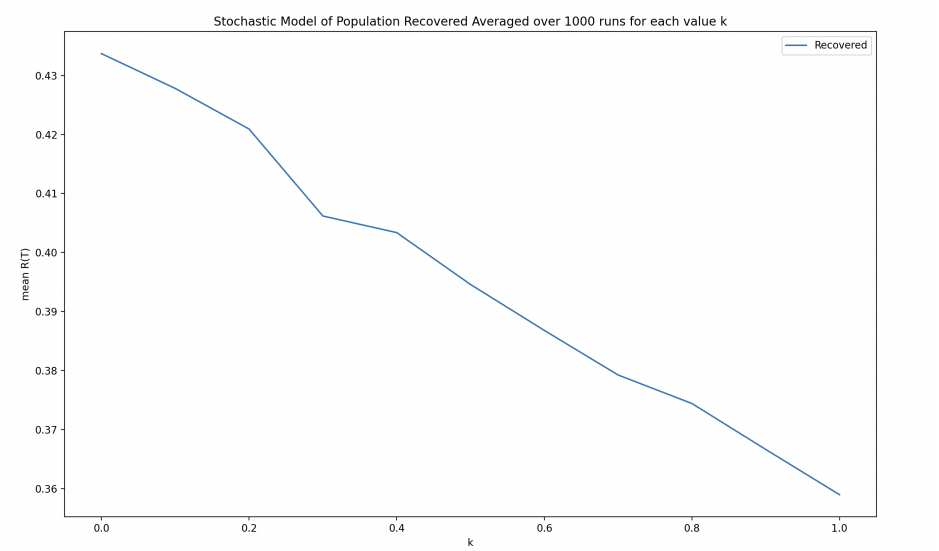

Figure 4: Q2.2 Plot

Assume you are a policymaker trying to reduce $R(T)$ by setting the proportion k of the population to get vaccinated. However, you do not have access to the ODE model above. Instead, you have an oracle that allows you to input the value of k and it outputs the value $R(T)$ after one simulation. Moreover, you are not just optimizing for $R(T)$. In fact, your cost function is:

$$Cost(k) = 8 \times \underbrace{(S_2(0) + I_2(0))}_{\text{Total fraction vaccinated}} + 10 \times R(T). \tag{2}$$

Since simulating the black box oracle is very costly, you choose to use a multi-arm bandit setup to kind the optimal k to minimize the total cost. You have 10 candidate arms/choices are $k \in \mathcal{K} = [0\%, 10\%, 20\%, \ldots, 90\%]$.

We will play the role of both the policy maker and the oracle to study the efficacy of the multi-arm bandit approach. We will assume the Oracle model is the ODE model discussed above with parameters sampled from distributions $\beta \sim Uniform(0.05, 0.15)$, $\gamma \sim Uniform(0.005, 0.015)$ and $\epsilon \sim Uniform(0.1, 0.3)$.

Q 2.3 (4 points) Using your implementation of the ODE model in Q2.2, write a function that samples the cost given the arm number as input. Specifically complete `cost_function` function. Assume that 90% of the population is susceptible and the rest are infected.

Solution: Code is located hw4/Q2/Q2.3.py

Multi-arm bandit setup:

As a policy maker, we do not know the effectiveness of choosing each of the arms in $\mathcal{K}$. Therefore, we will have an estimate of the cost as $V(k)$ which we will update after

each trial. During each trial t we will choose an arm based on a strategy π. The oracle will then simulate the ODE model with chosen arm $k(t)$ and provide the cost $c(t)$ as the output. Using the cost $c(t)$, the policy-maker will update the estimate of the cost $V(k(t))$ for arm $k(t)$. This setup is implemented in `run_bandit` function.

Q 2.4 (20 points) **Multi-arm bandit strategies:** We will use following two MAB strategies

1. ϵ-`GREEDY`: For each trial, we will choose a random arm with probability ϵ. Otherwise, we will choose the arm with minimum estimated cost $\arg\min_k V(k)$ where V is estimated from past trials.
2. `SOFTMAX`: For each trial, we choose arm k with probability $\frac{\exp(-V(k)/\tau)}{\sum_{k' \in \mathcal{K}} \exp(-V(k')/\tau)}$ where τ is the temperature.

Implement both ϵ-`GREEDY` and `SOFTMAX` policy. Specifically, complete the functions `epsilon_greedy` and `softmax`.

Set $\epsilon = 0.1$ for ϵ-`GREEDY` and $\tau = 1$ for `SOFTMAX` strategies. A single run of the MAB algorithm consists of running `run_bandit` for 100 trials (set `max_time = 100` in `run_bandit`). Perform 1000 independent runs of MAB for each of the two strategies and plot the average cost output by the oracle for each of the 200 trials i.e., run `run_bandit` for 1000 runs and average the output cost over 1000 runs. Submit a plot with the x-axis being 1-100 time-steps of running MAB and the y-axis being the average cost (averaged over 1000 runs) Which strategy performed better?

Note: This may take over 20 minutes to complete depending on code efficiency and compute resources.

Solution:

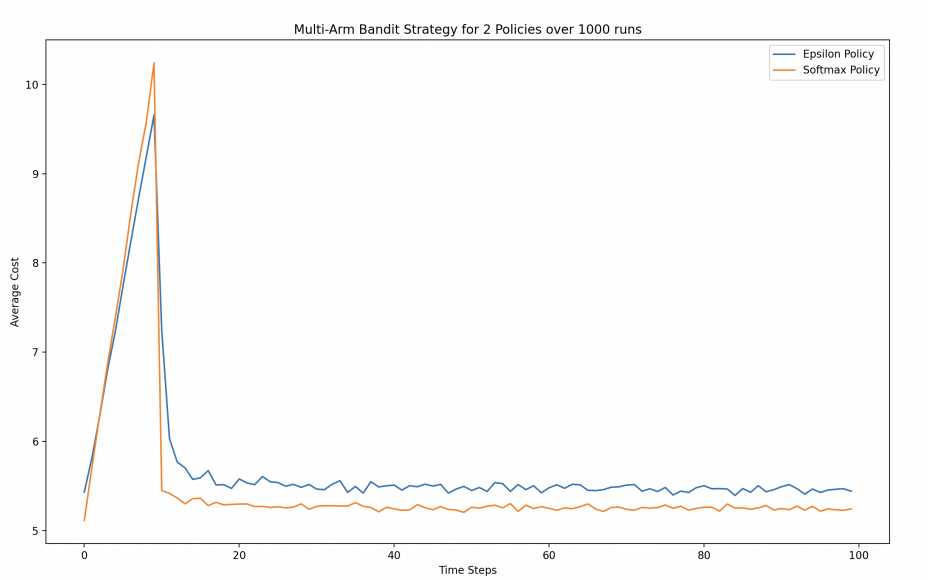

Figure 5: Q2.4 Plot

It seems as the softmax policy performs better over the long run since it has a lower cost. This makes sense because the epsilon does have a random element to it that may cause an increase in its cost.

Q 2.5 (4 points) Plot the average values of the cost estimate function V for both strategies.

How does it compare to the cost computed from average $R(T)$ calculated in Q2.2?

Solution:

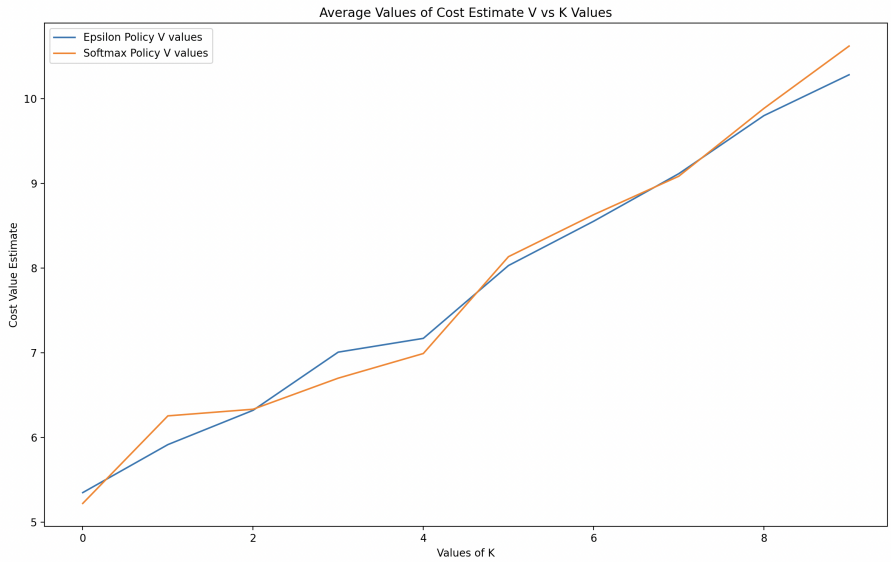

Figure 6: Q2.5 Plot

It seems that the average cost associated with the MAB algorithm are lower in comparison to the estimate cost values presented in the plot above. The cost value estimates seem to increase as k increases (maximum reaching around 11); however, the average cost from the average $R(T)$ from 2.2 is to hover between 5 and 6.

3. (40 points) Forecasting

Let's try to build a couple of simple ensemble models for forecasting. We have uploaded a csv file to Canvas. It shows the COVID-19 mortality, cases and a couple of auxiliary signals (mobility and testing) for the US national level at a weekly level in 2020. Our goal is to compare two ensemble models on how well they predict mortality for the month of September 2020 (epiweek 202036 to 202039), after training them on data from Mar-Aug 2020 (epiweek 202010 to 202035).

Q 3.1 (15 points) Create an ARIMA (2, 0, 2) model to forecast mortality. You will need to do rolling predictions (i.e., start with the training set (Mar-Aug 2020), create the ARIMA model, use it to predict one week ahead, then add the prediction to your training set, retrain, and then predict next week and so on. For example, you first use epiweek 202010 to 202035 as the training data to forecast 202036 mortality, and then use both epiweek 202010 to 202035 and your forecasted 202036 as the new training data to forecast 202037 mortality). Report the average RMSE error between your prediction and ground truth for epiweek 202036 to 202039 in the report PDF.

Note 1: You can use an off-the-shelf implementation of ARIMA. For python, we recommend using `statsmodel` package (Like: `from statsmodels.tsa.arima_model import ARIMA`[1].

[1]Here is a link explaining how to fit an ARIMA model and also how to do rolling predictions: `https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/`

Solution:
Epiweek 2020326 RMSE: 715.2633500079119
Epiweek 2020327 RMSE: 2065.9074207988087
Epiweek 2020328 RMSE: 2297.0784463330137
Epiweek 2020329 RMSE: 3222.2231147818093

Average RMSE across all 4 weeks: 2075.118082980386

Q 3.2 (10 points) Create a simple linear regression model, which takes in the number of cases, mobility, testing from a week and predicts mortality for the next week. *Note:* You can the OLS model in statsmodel (`from statsmodels.regression.linear_model import OLS`). Repeat what you did in Q3.1 for rolling predictions, and report the average RMSE error between your prediction and ground truth for epiweek 202036 to 202039 in report PDF.

Solution:
Epiweek 2020326 RMSE: 414.698823
Epiweek 2020327 RMSE: 466.131714
Epiweek 2020328 RMSE: 1350.700477
Epiweek 2020329 RMSE: 3695.904804

Average RMSE across all 4 weeks: 1481.85895442032

Q 3.3 (10 points) Now create two ensemble models EM1 and EM2. EM1 is just the average of your ARIMA and OLS models i.e.

$$EM1 = \frac{ARIMA + OLS}{2}.$$

EM2 is a weighted average of your ARIMA and OLS models i.e.

$$EM2 = \frac{w_1 \times ARIMA + w_2 \times OLS}{w_1 + w_2}.$$

Make the weight of each model in the ensemble equal to its (1/RMSE) error on the training set (epiweek 202010 to epiweek 202035), i.e., $w_1 = \frac{1}{RMSE_{ARIMA}}$ and $w_2 = \frac{1}{RMSE_{OLS}}$. Measure EM1 and EM2's average RMSE on epiweek 202036 to epiweek 202039 and report the RMSE error for epiweek 202036 to 202039 in the report PDF.

Solution:
Average RMSE across all 4 weeks for EM1: 2074.4701381576383
Average RMSE across all 4 weeks for EM2: 2065.997542233514

Q 3.4 (5 points) What do you observe comparing the test performance of your 4 models: ARIMA, OLS, EM1, EM2? Comment and try to explain the performance you observe in 1-2 lines in the report PDF.

Solution: With all 4 models, it seems ARIMA and EM1 performed the worst. OLS produced the lowest rmse, and EM2 was close to ARIMA and EM1's performance.

4. (8 points) Ethical and Societal Issues

Choose any one of the many facets of data science in epidemiology discussed in class (e.g. forecasting, surveillance, modeling, interventions, data collection, etc) and discuss various societal challenges associated with it such as ethics, privacy, anonymity, consent, equity, etc. Submit a short 500-word essay. This is an open-ended question, therefore feel free to read various resources and formulate your own points. Make sure to cite relevant works when you are making any factual claims.

Solution:
Epidemiological forecasting provides invaluable insight into disease trajectory to help epidemiologists and policymakers make influential decisions for the general population. However, forecasting requires vast data to give accurate insights into how the disease will progress. This is problematic because of the societal challenges that exist when utilizing hospital records, patient data, or surveillance data.

Data privacy plays a vital role in this societal challenge. Hospital records and patient data are sensitive pieces of information that may reveal critical health conditions that people may not want to be released. Hence, privacy laws keep this type of data anonymous or inaccessible. Despite this, an individual's health records can be rerouted to that individual through other publicly available data and complex data analysis techniques (Sweeney, 2000). Epidemiological forecasters need to ensure this data remains private through encryption or other means; nevertheless, privacy within data is always hard to maintain, and forecasters need to take extra steps to ensure the data remains private.

Informed consent can also pose another problem. Data collection, analysis, and forecasting must be done as fast as possible during pandemics and disease outbreaks to minimize risk to citizens. However, data revolving around patients may require informed consent, which can be a process that can take up valuable time during outbreaks. During the COVID-19 pandemic, government facilities and researchers collected vast data from data-tracing phone apps that raised privacy concerns. Since the government could not waste time since the sudden and deadly outbreak, they had no choice but to move past the informed consent process of obtaining an individual's data. However, this raised concerns about violating each individual's privacy (Morley et al., 2020).

Equity can pose a problem when it comes to individuals in underrepresented populations. Forecasting predictions may not accurately assess underrepresented communities if the data is biased by leaving out underrepresented populations. This can be detrimental since a policy that affects all individuals may not be the best for those minority communities. Hence, data collection is a key process during forecasting, and forecasters must ensure that the data is collected from various communities. However, sometimes, the data itself may not be available. In populations that do not have easy access to healthcare, hospital records may not be able to capture those populations in the first place. This scenario can exacerbate existing healthcare disparities in underdeveloped communities as the policy may need the proper resources or mandates to help them.

Lastly, forecasting data should be conveyed to the population as is. Sometimes, news networks and public health agencies can raise unnecessary alarms for the population when forecasting results come out. However, there are always limitations to these types of data, and these limitations should always be conveyed to the general population. News networks may sometimes stretch the truth regarding these forecasts to increase public viewership; however, this can cause public panic, and it should be avoided as it does not help the already present problem of the outbreak.

These are all valid concerns when it comes to epidemiological forecasting. It's important to recgonize these concerns as data scientists and take steps now to potentially fix these problems.

Citations:

Flaherty, H., et al. (2021). Data privacy challenges in public health forecasting. Journal of Public Health Data.

Sweeney, L. (2000). Simple demographics often identify people uniquely. Carnegie Mellon University.