Homework 4: Forecasting, Interventions, Immunization

Released: Nov 3, 2022; Due: **5pm ET**, Nov 15, 2022(1 Late day)

Georgia Tech	CSE 8803 EPI Fall 2022
College of Computing	**Student NAME:** Anshuman Sinha
B. Aditya Prakash	**Student GTID:** 903843345

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

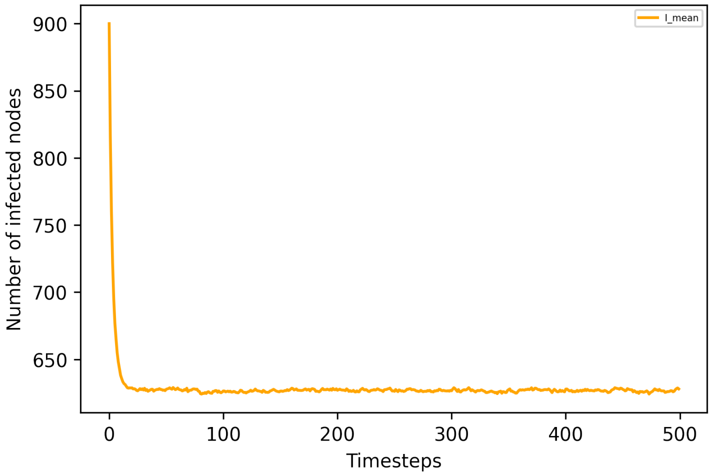
Consider the so-called acquaintance immunization policy (where we pick a uniformly random neighbor of a uniformly random node) and the uniformly random strategy (of picking a node at random). For each policy, you should keep sampling till you have picked k different nodes (where k is the budget). Let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

Generate a undirected unweighted graphs: 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 (5 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:

The average number of infected nodes at each step till max_time averged over 200 iterations.



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: Nodes chosen according to each policy:

RANDOM: The following is the array I got for friends: array after setting : np.random.RandomState(0)

array([993, 859, 298, 553, 672, 971, 27, 231, 306, 706, 496, 558, 784, 239, 578, 55, 906, 175, 14, 77, 31, 481, 310, 311, 883, 788, 45, 103, 760, 1, 823, 710, 614, 790, 408, 736, 957, 366, 918, 267, 230, 996, 635, 698, 251, 783, 819, 141, 316, 587, 331, 295, 262, 432, 862, 582, 272, 270, 987, 319, 569, 643, 142, 202, 413, 196, 264, 531, 252, 576, 738, 299, 740, 247, 926, 412, 389, 796, 601, 654, 261, 456, 386, 982, 909, 693, 236, 501, 497, 874, 452, 494, 923, 279, 638, 485, 568, 108, 367, 644])

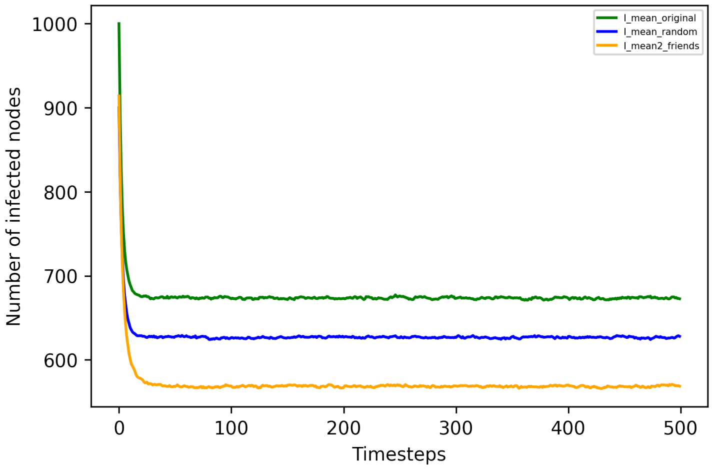
FRIENDS: The following is the array I got for friends: array after setting : np.random.RandomState(0)

array([619, 682, 33, 121, 2, 92, 356, 550, 264, 461, 144, 5, 136, 342, 543, 389, 69, 615, 4, 245, 9, 342, 697, 678, 375, 27, 13, 252, 545, 184, 83, 378, 7, 5, 176, 679, 630, 31, 932, 32, 53, 53, 196, 2, 6, 26, 406, 37, 52, 5, 142, 191, 567, 6, 0, 68, 142, 2, 11, 210, 33, 480, 0, 176, 663, 635, 306, 54, 103, 19, 199, 12, 728, 198, 699, 302, 268, 7, 58, 226, 7, 168, 728, 563, 76, 235, 9, 44, 29, 57, 260, 283, 933, 93, 158, 489, 854, 407, 47, 249])

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 Q1.3). Remove these nodes from the G_{ba}, and re-run the SIS model on the new (smaller) versions of each graph. Generate a plots: plot the average number of infections vs time when (a) no nodes have been removed (b) when nodes have been removed according to FRINEDS 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:

The average number of infected nodes for each strategy at each step till max_time averged over 50 iterations.



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:

- We observe that after removing the friends node we get a lower final (saturated) level of infections.
- These are due to the better connected nodes getting immunised, hence although the disease will stay forever but will be slow in spreading.
- Which means people will have more time to recover (and become susceptible), and hence the value of I will saturate to a lower value.

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 setup a variant of SIR model that accounts for influence of vaccination rate with disease spread. Next we will setup the problem of choosing an optimal vaccination strategy for our SIR model. Finally, we will use various multi-arm bandit stratgies to solve for optimal vaccination strategy.

We will simulate a variation of 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 fraction of total population that are susceptible, infected and recovered as well as are unvaccinated. Similarly, let $S_2(t), I_2(t)$ and $R_2(t)$ be fraction of population that susceptible, infected and recovered as well as are 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 effectiveness of the vaccine in both reducing 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.

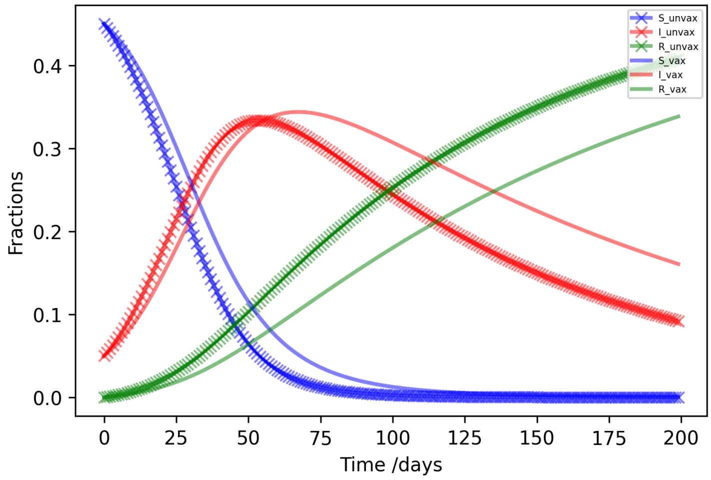
Q 2.1 (6 points) Implement the above defined SIR model and submit the code. Specifically complete the `model_ode` function in the notebook.

Let 95% 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 population 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 $k = 10\%$ and 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?

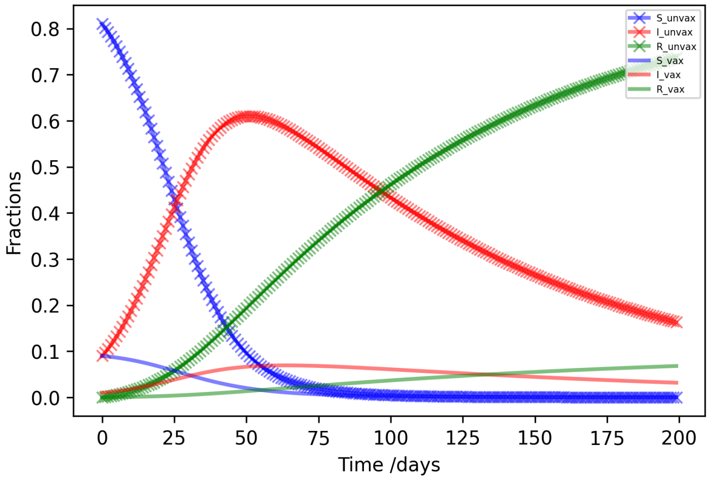
Hint: Refer to Q1 in HW 1.

Solution:

The fractions of infected nodes for each strategy at each step till max_time with k = 50 %



The fractions of infected nodes for each strategy at each step till max_time with k = 10 %



We observe a larger Infected population when vaccination is lowered, which means larger recovered poplations as well.

The final R values for 50 percent and 10 percent vaccination is as follows:

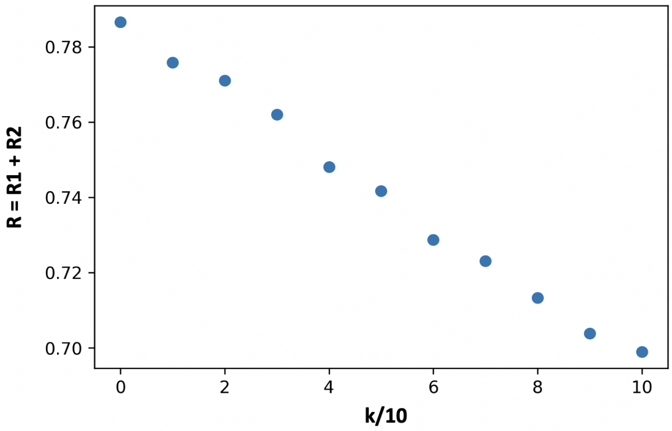
- R1_50% = 0.7463228199286038
- R1_10% = 0.8047133333921389

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, sampling 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 eack values of k).

Solution:

The average (over 1000 runs) R(T) for each k %



Assume you are a policy maker trying to reduce $R(T)$ by setting the proportion k of the population to get vaccinated. However, you don not have access to the ODE model above. Instead you have a 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 policy maker and the oracle to study the efficacy of 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 are susceptible anf rest are infected.

Solution:

Done in code section.

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.

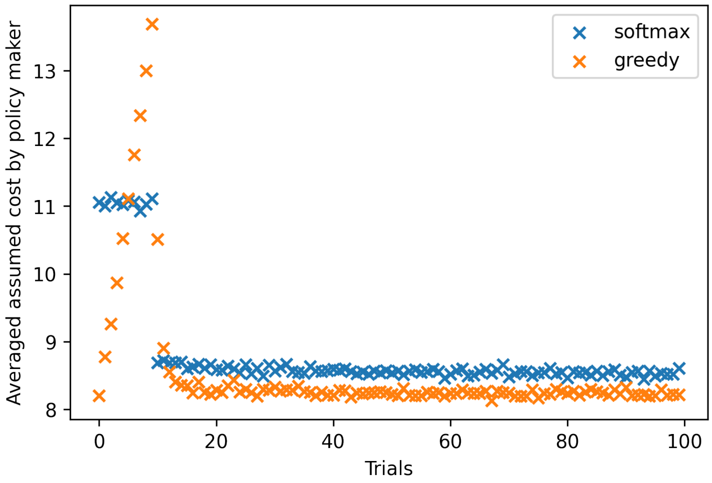
Implement both ϵ-`GREEDY` and `SOFTMAX` policy. Specifically complete the functions `epsilon_greedy` and `softmax`.

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 x-axis being 1-100 time-steps of running MAB and 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:

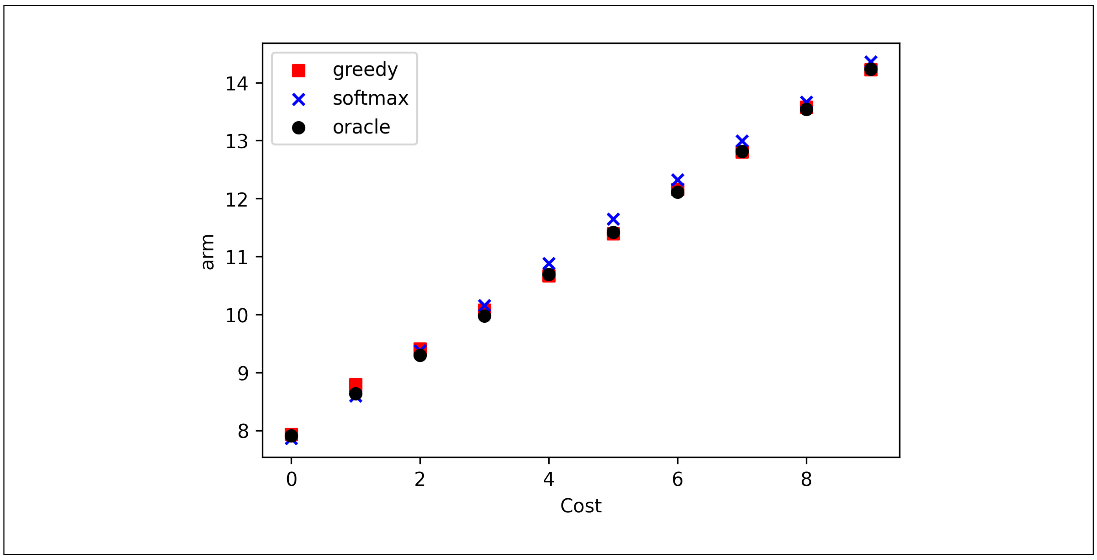
Plot of average cost of each of the 2 strategies. We see that the Greedy approach performs better as the final cost turns out to be lower that the softmax, also they saturate at similar times.



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:

Plot the average values of the cost estimate function V for both strategies and average cost of R(T). They have a good match over the various arms, although the softmax tens to overpredict a little!



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 a off the shelf implementation of ARIMA. For python, we recommend using `statsmodel` package (Like: `from statsmodels.tsa.arima_model import ARIMA`[1].

Solution:
RMSE error = 1984.330085840417

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 Q2.1 for rolling predictions, and report the average RMSE error between your prediction and ground truth for epiweek 202036 to 202039 in report PDF.

[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:
RMSE 2 = 1992.0691496084992

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:
RMSE EM1 = 1722.7403416342659 RMSE EM2 = 1742.8553618689239

Q 3.4 (2 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:

- In general it is difficult to compare the 4 models as the errors are within similar range. Although one can comment upon the formulation, The ARIMA model should better perform than the OLS as, the ARIMA model may account for non-linearity in the data , but the OLS model is a linear one.
- Moreover, between EM1 and EM2 , the EM1 model is just averaging the results, whereas EM2 is taking a weighted average. With better weights for the better predicting model! Hence EM2 should in general perform better than EM1.
- Further, EM2 should be better than ARIMA, in sense that. EM2 will have better control over curvature prediction.

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 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 you own points. Make sure to cite relevant works when you are making any factual claims.

Solution:

Ethical and Societal Issues in epidemiological data collection

Planning, implementing, and evaluating public health practices require the ongoing, systematic collection, analysis, and interpretation of health-related data. These datasets form the backbone of any computational epidemiological model, in predicting and forecasting disease spreads.

The epidemiological data collection and examination are likely to have significant positive social effects, it is crucial to be aware of and prepare for potential hazards and unforeseen consequences.Ethical blunders or misinterpretations may result in social rejection and/or skewed laws and regulations, which in turn hinder the acceptance and development of [the essential] data science.

To avoid the release of individually identifiable location data, they also pose security and privacy concerns [4]. Data collection from satellites is also constricted to different extents by privacy and security laws in different regions that may introduce irregularity in data collection.[5]

Data source like contact data can also have several difficulties [6, 7]. First, location data obtained by contact tracking apps raises privacy issues, especially when centralized methods are used. Using surveillance footage or collecting financial transactions to track someone involuntarily raises even more privacy and security issues. Second, there isn't widespread use of the majority of voluntary contact tracing techniques. For instance, less than 20 percent of people used the government-sponsored TraceTogether app. Due to Bluetooth and GPS's errors, there are additional issues with location measurements [8]. Additionally, it should be remembered that introducing contact tracing could occasionally have negative outcomes. For instance, it can make it harder for people to follow the rules set up to stop the epidemic [9].

In datasets containing sensitive information, there are additional issues that concern privacy implications (e.g., EHRs). We might take use of developments in the quickly developing disciplines of differential privacy and federated learning [1], as well as promote discussions on the ethics of privacy and fairness while using this data [2]. Finally, I wish to point the significance of initiatives like [3] that aim to create open data archives that allow researchers to access various data versions. Research on the aforementioned data quality challenges can go more quickly if efforts are made to create infrastructures that make it simple to access data archives from many sources.

1. Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. Knowledge-Based Systems 216 (2021), 106775.
2. Thilo Hagendorff. 2020. The ethics of AI ethics: An evaluation of guidelines. Minds and Machines 30, 1 (2020), 99–120
3. Alex Reinhart, Logan Brooks, Maria Jahja, Aaron Rumack, Jingjing Tang, Sumit Agrawal, Wael Al Saeed, Taylor Arnold, Amartya Basu, Jacob Bien, et al. 2021. An open repository of real-time COVID-19 indicators. Proceedings of the National Academy of Sciences 118, 51 (2021).
4. John Krumm. 2009. A survey of computational location privacy. Personal and Ubiquitous Computing 13, 6 (2009), 391–399.

5. Patrick Butler, Naren Ramakrishnan, Elaine O Nsoesie, and John S Brownstein. 2014. Satellite imagery analysis: What can hospital parking lots tell us about a disease outbreak? IEEE Annals of the History of Computing 47, 04 (2014), 94–97.
6. Jinfeng Li and Xinyi Guo. 2020. COVID-19 contact-tracing apps: A survey on the global deployment and challenges. arXiv preprint arXiv:2005.03599 (2020).
7. Krista C Swanson, Chiara Altare, Chea Sanford Wesseh, Tolbert Nyenswah, Tashrik Ahmed, Nir Eyal, Esther L Hamblion, Justin Lessler, David H Peters, and Mathias Altmann. 2018. Contact tracing
8. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 1724–1734
9. Robert A Kleinman and Colin Merkel. 2020. Digital contact tracing for COVID-19. CMAJ 192, 24 (2020),E653–E656.