Homework 3: Detection, Surveillance

Released: Oct 7, 2025; Due: **5pm ET**, Oct 21, 2025

Georgia Tech
College of Computing
B. Aditya Prakash

CSE 8803 EPI Fall 2025
Student NAME: XXXXXXXX
Student GTID: XXXXXXXX

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. (15 points) Submodularity

Q 1.1 (5 points) Prove that the coverage function from our lecture is a monotone submodular function. Recall that the coverage function takes a collection of sets $\{S_i\}_{i=1}^n$ and outputs the size of their union $|S_1 \cup S_2 \cup \cdots \cup S_n|$.

Solution:

Q 1.2 (5 points) Let f be a monotone submodular function. Show that for any two sets S and T: $f(T) - f(S) \leq \sum_{e \in T} (f(S + e) - f(S))$.

Solution:

Q 1.3 (5 points) Let f be a monotone submodular function. Show that the following function is also submodular: $h(A) = \min(f(A), f(V/A))$ where V is the universal set of all elements.

Solution:

Q 1.4 (5 points) **[Bonus]** Let f be a monotone submodular function and let g be a concave function. Show that the following function is also submodular: $h(A) = g(f(A))$.

Solution:

2. (30 points) Subset Scan and Anomaly Detection

We will try to implement a simple anomaly detection algorithm for detecting outbreaks using ideas similar to the subset scan methods we saw in class. For this question, we will use the data from the county-level influenza-like infections across counties in New York [1].

As a beginning, extract these columns from the raw data table: 'Week Ending Date', which is the end day of each included weeks; 'County', which is the county where the infections are collected; 'Disease', which corresponds to one of the three: 'INFLUENZA_A', 'INFLUENZA_B', and 'INFLUENZA_UNSPECIFIED'; 'Count', which is the collected infection counts. In this problem, you should only consider the infection counts of 'INFLUENZA_A'.

Q 2.1 (5 points) We start with a rough visualization.

We hope you to first cumulate the INFLUENZA_A infections of each county across all the time, find the 5 counties with largest cumulated infections, and plot their weekly infections in one plot. (That is, you should plot the weekly counts of each county in one graph. The label of each plotted sequence should be the name of county – remember to show the legends).

Based on this visualization, did you see any obvious anomalies? Explain your finding.

Solution:

Q 2.2 (10 points) After a visualization, we now come to quantitized anomaly detection. As a beginning, use all the weeks ended before 2017-01-01 as the training set (that means, you should calculate all the parameters based on this subset). Calculate the mean and stds of the INFLUENZA_A infections of each county in the train set, and report them here.

You should submit in this format: 'Means: COUNTY1: x.xx COUNTY2: x.xx ... Stds: COUNTY1: x.xx COUNTY2: x.xx ...'

(The counties should be printed in an alphabetical order)

Solution:

Q 2.3 (5 points) In statistics, you may have heard about the '3 sigma' rule for anomalies i.e., anything beyond 3 standard deviations from the mean is unlikely and hence an anomaly [2]. Hence, use the mean and std-deviation you get in Q2.2 for each week in testing set (i.e. all weeks that ends on or after 2017-01-01) and apply this rule to compute how many weeks in the test set will be marked anomalous. Report the 10 counties with the most anomaly weeks, and their corresponding anomaly week counts. (If multiple counties share the same anomaly counts, break the tie by lower alphabetical order first)

Solution:

Q 2.4 (10 points) Now, we wish you to achieve subset scan results considering spatial relationship. Specifically, you should refer to the county_adjacency2025.txt (which is also used in earlier HWs). For each of the county, you should find all the adjacent counties, and treat the average of the weekly infections across all the adjacent counties together with that county itself as an alternative weekly infection of that county. After that,

[1] https://health.data.ny.gov/Health/Influenza-Laboratory-Confirmed-Cases-by-County-Beg/jr8b-6gh6/about_data

[2] https://en.wikipedia.org/wiki/68%E2%80%9395%E2%80%9399.7_rule

calculate the mean and variance again on all weeks ended before 2017-01-01 and report the ten counties with the most anomaly weeks ended after 2017-01-01. Did you see any change from this result and the result in Q2.3? Explain possible reasons.

Solution:

3. (30 points) Sensors for detection in Network Model

We are going to empirically look at various strategies for selecting social network sensors to detect epidemic outbreaks. We will use the graph in `facebook.txt` [3] which contains a small subset of friendship network of a Facebook group. We will use the SI model with $\beta = 0.005$ to simulate the infection.

Q 3.1 (4 points) As a warmup, simulate the SI model for up to $T = 100$ time-steps for 100 runs and plot the average fraction of S, I vs t curve. Select 4 nodes at random to be infected at $t = 0$. Also plot the average number of daily infected people for $t = 0, \ldots, 100$. Report the average time t^* at which the maximum daily infections are maximum.

Hint: Use the implementation of SI model in `sis_model.py` file. If you use another language you can convert the logic of the code in the file.

Solution:

Q 3.2 (10 points) Since the graphs are large, it is not always feasible to keep track of the states of all nodes in practice. Therefore, we will select a smaller subset of nodes to track during the epidemic which we call sensors. We will implement three strategies for sensor selection:

- `RANDOM`: We choose k nodes uniformly at from the graph
- `FRIENDS`: We choose k nodes uniformly at random and for each of them we select a random friend. We will use these friends as the sensors.
- `CENTRAL`: We select the top k nodes with largest *eigenvector centrality* [4]. You can use functions like `nx.eigenvector_centrality_numpy`.

Set $k = 100$ for all strategies. The file `rand_nodes.npy` contains a random list of nodes. Select the first k nodes from the list to implement the `RANDOM` strategy. For `FRIENDS` strategy, use the k selected nodes from `RANDOM` strategy to randomly select their friends. Simulate the SI model for $T = 100$ steps and note the fraction $\tilde{I}(t)$ of the sensors that are infected at each time step for each strategy. Note that sensors can also be infected at $t = 0$.

Plot average $\tilde{I}(t)$ vs t for all three strategies averaged over 20 runs. Also plot the average number of daily infections $\tilde{I}_d(t) = \tilde{I}(t) - \tilde{I}(t-1)$ over time.

Solution:

Q 3.3 (6 points) Report the peak time $\tilde{t}^*$ and peak daily infection for all 3 strategies (the time t where $\tilde{I}_d(t)$ is maximum is peak time).

The value $t^* - \tilde{t}^*$ is the *lead time* i.e., the difference in time between the detection of epidemic peak among sensors and the time when it peaks in the entire population. Report the lead time for all 3 strategies.

[3] taken from `https://snap.stanford.edu/data/ego-Facebook.html`

[4] `https://en.wikipedia.org/wiki/Eigenvector_centrality`

Solution:

Q 3.4 (4 points) Compare the lead-time of various strategies and explain the difference in lead-time of various strategies.

Solution:

Q 3.5 (6 points) Now repeat Q 3.2 for $k = 50$ and $k = 500$. For each strategy submit a $\tilde{I}(t)$ vs t plot comparing different values of k. How does the lead time change with a value of k for each strategy?

Solution:

4. (25 points) Flu Surveillance using Google Symptoms Data

We will study the efficacy of Google Symptoms Data [5] as a source of Flu surveillance. We measure the usefulness of a signal using the correlation with ILI signals collected by CDC [6]. Specifically, we will look at the 2018-19 season (datasets can be downloaded from Canvas).

Q 4.1 (2 points) We will start by considering the state of Georgia. Extract the `% Unweighted ILI` from `ILINet_states.csv`. Plot the weekly `Unweighted ILI` of Georgia over the 2018-19 flu season. (Refer to Q2 for details on flu seasons.)

Solution:

Q 4.2 (10 points) CDC lists various symptoms for Flu [7]. We will consider the following symptoms: `Fever, Low-grade fever, Cough, Sore throat, Headache, Fatigue, Muscle weakness`.

Extract the Symptoms trends for each of these sympotms from the files `2018_symptoms_dataset.csv` and `2019_symptoms_dataset.csv` over the weeks of 2018-19 seasons. Submit a single plot showing the trends of all the symptoms over the 2018-19 seasons with the x-axis showing the Epiweeks and the y-axis the symptom trend values.

Solution:

Q 4.3 (3 points) We will use Pearson Correlation Coefficient (PCC) [8] to measure how correlated each of symptom's trend is to ILI. Evaluate PCC for each of the symptoms with ILI for the 2018-19 season in Georgia. You may use functions like `scipy.stats.pearsonr` to evaluate PCC. Also, submit the code.

Solution:

Q 4.4 (10 points) Repeat Q4.1 and Q4.2 for following states: `California, Texas, New York, Alaska` and `Mississippi`. Each state including Georgia, also reports the symptom with the highest PCC along with the value of PCC. Can you think of any reasons for the differences in PCC across these states?

[5] `https://pair-code.github.io/covid19_symptom_dataset/`
[6] `https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html`
[7] `https://www.cdc.gov/flu/symptoms/symptoms.htm`
[8] `https://en.wikipedia.org/wiki/Pearson_correlation_coefficient`

Solution:

Q 4.5 (5 points) **[Bonus]** There may be some lead-time between reported ILI and the symptoms trend. Here, we define the lead-time t_s of a symptom s to be the value of t' that maximizes the PCC between ILI time-series from week $t' + 1$ to T and time-series of symptom s from 1 to $T - t'$ where $T = 52$ week. Intuitively, we measure the delay between the symptoms signal and the ILI signal.

For the most correlated symptoms you calculated in Q4.3 for each of the 6 states, find the lead time for the respective symptom. Submit the code.

Solution: