ELEC 677: Network Science and Analytics **Fall 2018**

Homework 3

Hand-out Date: 10/18/18 **Due Date: 11/01/18**

This homework contains only two exercises. The first exercise is shorter, and you will explore some of the relations that we discussed in class between partial correlations and Gaussian graphical models when studying association networks. In the second exercise, you will study and simulate an epidemic spread in a real-world network.
The grading is over 100 points distributed as indicated throughout the document.

3.1) Partial correlations and Gaussian graphical models [16 points]. For $S_m = \{k_1, \ldots, k_m\}$, we defined the partial correlation of node attributes X_i and X_j adjusting for $\mathbf{X}_{S_m} = [X_{k_1}, \ldots, X_{k_m}]^\top$, as

$$\rho_{ij|S_m} = \frac{\sigma_{ij|S_m}}{\sqrt{\sigma_{ii|S_m}\sigma_{jj|S_m}}}. \tag{3.1}$$

In the above definition, $\sigma_{ii|S_m}$, $\sigma_{jj|S_m}$, and $\sigma_{ij|S_m}$ are the diagonal and off-diagonal elements of the 2×2 partial covariance matrix

$$\mathbf{\Sigma}_{11|2} = \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}, \tag{3.2}$$

where $\mathbf{\Sigma}_{11}$, $\mathbf{\Sigma}_{22}$, and $\mathbf{\Sigma}_{12} = \mathbf{\Sigma}_{21}^\top$ are defined through the partitioned covariance matrix

$$\text{cov}\begin{bmatrix}\mathbf{W}_1 \\ \mathbf{W}_2\end{bmatrix} = \begin{pmatrix}\mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22}\end{pmatrix}, \qquad \text{where } \mathbf{W}_1 = [X_1, X_2]^\top \text{ and } \mathbf{W}_2 = \mathbf{X}_{S_m}. \tag{3.3}$$

Note that for $m = 0$, partial correlations reduce to standard Pearson correlations.

For the partial correlations in (3.1), there exist recursive relationships among coefficients at adjacent orders $m-1$ and m that allow for their efficient calculation. In the first part of this problem you will derive this relationship for the case of $m = 1$ and three random variables, say X, Y and Z. Specifically, you are asked to derive an expression relating $\rho_{XY|Z}$ to the Pearson correlations ρ_{XY}, ρ_{XZ}, and ρ_{YZ}. Throughout this whole exercise we assume that X, Y, and Z have unit variance.
a) *[5 points]* Use (3.2) to derive a closed-form expression for the partial covariance matrix $\mathbf{\Sigma}_{XY|Z}$. Then follow (3.1) to combine the elements of the matrix that you just obtained to show that

$$\rho_{XY|Z} = \frac{\rho_{XY} - \rho_{XZ}\rho_{YZ}}{\sqrt{(1-\rho_{XZ}^2)(1-\rho_{YZ}^2)}}. \tag{3.4}$$

b) *[3 points]* Assume now that $[X, Y, Z]^\top$ is a multivariate Gaussian vector with zero mean and covariance $\mathbf{\Sigma}$. Consider the task of optimally predicting Z as a linear combination of X and Y, using mean-squared error (MSE) as criterion. That is, consider the MSE minimization problem

$$\min_{\beta_{ZX}, \beta_{ZY}} \mathbb{E}[(Z - \beta_{ZX}X - \beta_{ZY}Y)^2]. \tag{3.5}$$

Show that the MSE cost takes the form

$$1 + \beta_{ZX}^2 + \beta_{ZY}^2 - 2\beta_{ZX}\rho_{ZX} - 2\beta_{ZY}\rho_{ZY} + 2\beta_{ZX}\beta_{ZY}\rho_{XY}. \tag{3.6}$$

c) *[3 points]* Differentiate (3.6) to show that the optimal predictor $[\hat{\beta}_{ZX}, \hat{\beta}_{ZY}]^\top$ satisfies the following linear system of equations

$$\begin{bmatrix}\rho_{ZX} \\ \rho_{ZY}\end{bmatrix} = \begin{pmatrix}1 & \rho_{XY} \\ \rho_{XY} & 1\end{pmatrix}\begin{bmatrix}\hat{\beta}_{ZX} \\ \hat{\beta}_{ZY}\end{bmatrix}. \tag{3.7}$$

d) *[5 points]* Solve the above system of equations and compare its solution to the result that you showed in (3.4). Hence argue that $\hat{\beta}_{ZX} = 0$ if and only if $\rho_{ZX|Y} = 0$, and likewise for $\hat{\beta}_{ZY} = 0$.

3.2) Epidemics across the world [84 points]. In this problem we will consider the issue of epidemic spreading on networks and how to prevent their outbreaks. We will consider theoretical and computational aspects of an epidemic model while focusing on a real-world network of global flight connections as our main dataset.

a) *[20 points]* Construct a network from the flight data. You have been provided with 2 `csv` files (named `airport_Nodes_GC.csv` and `airport_Edges_GC.csv`) that contain all the information of all the airports (nodes), and flights (links)[1]. The edges are directed and weighted, and are proportional to the number of daily flights from one airport to another. You should construct a network G from this data as follows.

- Read in the edge data and create an initial (directed) network.
- We will work with undirected graphs, so form the undirected graph of this network by symmetrizing its adjacency matrix as follows
$$\mathbf{A}_{sym} = (\mathbf{A} + \mathbf{A}^\top)/2.$$
- If the undirected graph obtained is not connected, keep its giant component as your new graph and work with it from here onwards.

Hint: Note that command `networkx.DiGraph.to_undirected` returns an undirected graph with the same name and nodes and with edge $(u, v, data)$ if either $(u, v, data)$ or $(v, u, data)$ is in the digraph. If both edges exist in the digraph and their edge data is different, **only one edge is created with an arbitrary choice of which edge data** to use. You must check and correct for this manually if desired, or use other method to obtain the symmetry operation required above.

b) *[20 points]* Plot the airport network. Plot the flight network, using some of the extra information for the node files: use the longitude and latitude coordinates provided when drawing the nodes, and scale the node size according to their eigenvector centrality. You should take into account the weights of the edges when computing this centrality. You may find it useful as well to adjust the transparency of the edges for visual clarity. **Hint:** As a sanity check, the two nodes with highest eigenvector centrality (eigenvector normalized to have unit norm) should have the values approximately 0.179 and 0.170.

We will now study an epidemic model on this network, where every node is in one of two states: it is either infected, or susceptible to being infected. We denote the set of the vertices by $V = \{v_1, \ldots, v_n\}$ and the adjacency matrix by $\mathbf{A} = [a_{ij}]$. The state of node v_i at time $t \geq 0$ is a binary random variable $X_i(t) \in \{0, 1\}$. The state $X_i(t) = 0$ (resp., $X_i(t) = 1$) indicates that node v_i is in the susceptible (resp., infected) state. We define the vector of states as $X(t) = [X_1(t), \ldots, X_n(t)]^\top$. The state of a node can experience two possible stochastic transitions:

i) Assume node v_i is in the susceptible state at time t. This node can switch to the infected state during the time interval $[t, t + \Delta)$ with a probability that depends on: (a) an infection rate β, (b) the probability of contact with its neighboring nodes $N_i = \{j \mid a_{ij} \neq 0\}$ and their states. We can write the probability of this transition as

$$\text{Prob}[X_i(t + \Delta) = 1 | X_i(t) = 0, X(t)] = 1 - \Pi_{j \in N_i \wedge X_j(t) = 1}(1 - \beta a_{ij}). \tag{3.8}$$

We usually have $\beta \ll 1$. Therefore, instead of (3.8), we use the following first-order approximation:

$$\text{Prob}[X_i(t + \Delta) = 1 | X_i(t) = 0, X(t)] = \sum_{j \in N_i} \beta a_{ij} X_j(t). \tag{3.9}$$

ii) Assuming that node v_i is infected, the probability of v_i recovering back to the susceptible state in the time interval $[t, t + \Delta)$ is given by

$$\text{Prob}[X_i(t + \Delta) = 0 | X_i(t) = 1] = \gamma, \tag{3.10}$$

where $0 \leq \gamma \leq 1$ is the curing rate.

[1] The original dataset was made available by Tore Opsahl and is discussed in the blog post *Why Anchorage is not (that) important: Binary ties and Sample selection.* Available at `http://wp.me/poFcY-Vw`.

c) *[4 points]* Show that the first-order approximation in (3.9) is in fact an upper bound for the original transition probabilities in (3.8).

The spread model characterized by (3.9) and (3.10) may be hard to analyze for large-scale networks. One standard approach is to use a *mean-field approximation* of the model: define $p_i(t) = \text{Prob}[X_i(t) = 1] = \mathbb{E}[X_i(t)]$, i.e., the marginal probability of node v_i being infected at time t. We can use (3.9) and (3.10) to approximate the dynamics of $p_i(t)$:

$$\frac{dp_i(t)}{dt} = \beta(1 - p_i(t)) \sum_{j=1}^{n} a_{ij} p_j(t) - \gamma p_i(t). \tag{3.11}$$

This approximation is widely used in the field of epidemic analysis and control, since it performs numerically well for many realistic network topologies.
d) *[20 points]* Simulating the spread dynamics using the mean-field approximation. Suppose that the first 20 nodes (according to their ids in the dataset) are initially infected. Use (3.11) to simulate the evolution of the expected size of the infection defined as $\bar{p}(t) = \sum_{i=1}^{n} p_i(t)$ over time, assuming a recovery rate $\gamma = 0.1$ and an infection rate $\beta = 0.1$. (Hint: you can use $\frac{dp_i(t)}{dt} \approx \frac{p_i(t+\Delta) - p_i(t)}{\Delta}$ where $\Delta = 0.05$ and time horizon $[0, 5]$). Repeat the experiment for $\beta = 0.05$ and $\beta = 0.01$. Plot three curves (one for each value of β) showing the evolution of the expected size of the infection over time.

It can be shown that a sufficient condition for virus extinction, that is to ensure that the virus will eventually die out, is to have

$$\lambda_{\max}(\mathbf{A}) < \frac{\gamma}{\beta}. \tag{3.12}$$

An immunization strategy is to choose a subset of nodes $I \subseteq V$ and immunizing them against the virus. An immunized node can neither get infected nor pass the infection. We call an immunization strategy "effective" if it results in the eventual extinction of the virus (almost surely), *no matter how widespread the initial infection is*. A potential idea for designing an effective immunization strategy is to rank the nodes based on some centrality measure and immunize them in order of their centralities until the condition (3.12) is satisfied. Note that immunizing node v_i is equivalent to removing the i-th row and column from $\mathbf{A}$.
e) *[20 points]* Assume that $\beta = 0.01$ and $\gamma = 0.4$. (i) What is the minimum number of immunizations required to satisfy condition (3.12) if you use (weighted) degree centrality to sort the nodes? (ii) What is this minimum number when you use (weighted) eigenvector centrality instead?