Statistical analysis of network data lecture 12

Sofia Olhede

December 4, 2024

1. Link Prediction
2. Scoring methods
3. Biclustering

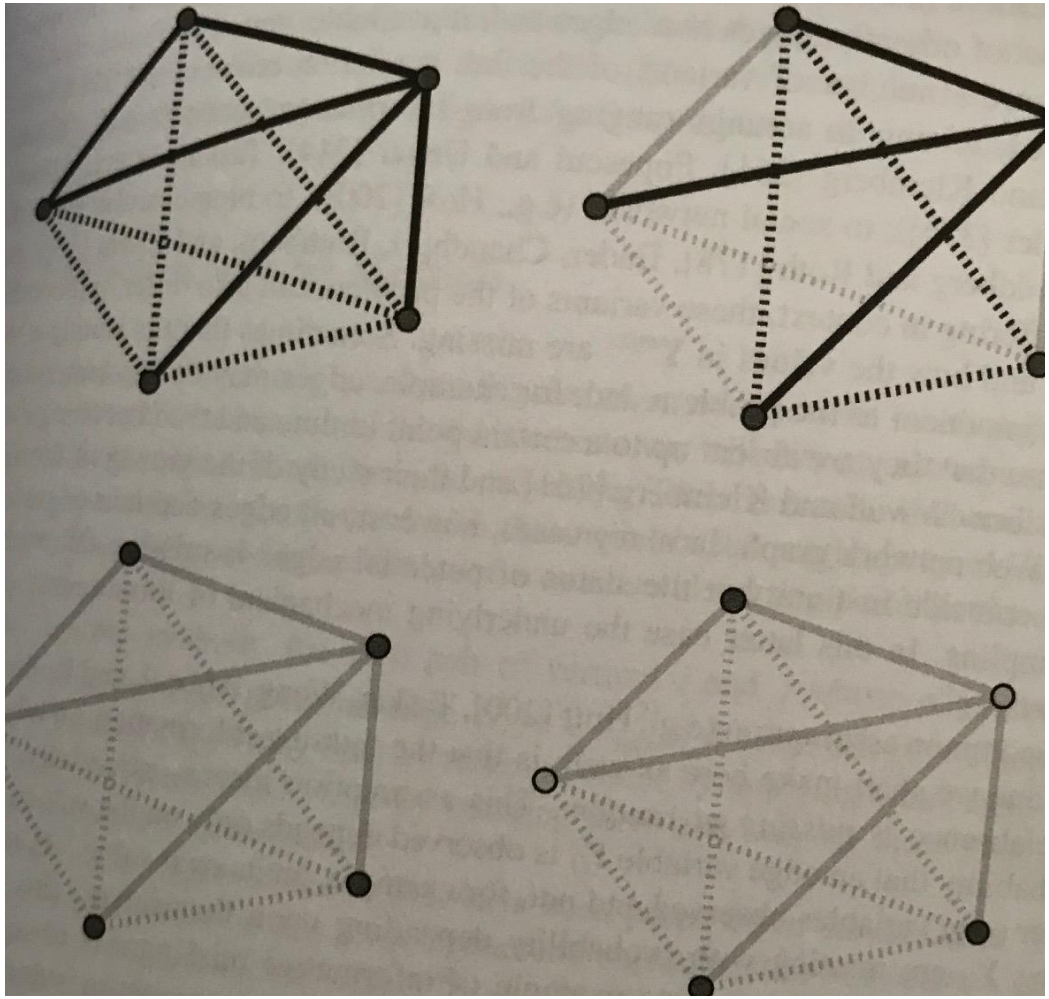
Link Prediction I

- Let $G = (V, E)$ be an undirected network.
- Assume that all vertices $v \in V$ are known.
- We let $V^{(2)}$ be the set of distinct unordered edges.
- Note that the union of (i) the set E of edges in G, and (ii) the set $V^{(2)} \setminus E$ of non-edges.
- In this section we will suppose the presence or absence of an edge in $V^{(2)}$ is only observed for $V^{(2)}_{\text{obs}}$.
- For $V^{(2)}_{\text{miss}} = V^{(2)} \setminus V^{(2)}_{\text{obs}}$ the information about edges is missing.

Link Prediction II

- Let $\mathbf{A}$ be a random $n \times n$ matrix with binary entries.
- We denote by $\mathbf{A}^{\text{obs}}$ and $\mathbf{A}^{\text{miss}}$ the adjacency matrices corresponding to $V^{(2)}_{\text{miss}}$ and $V^{(2)}_{\text{obs}}$.
- The problem of **link prediction** is to predict $\mathbf{A}^{\text{miss}}$ given $\mathbf{A}^{\text{obs}} = \mathbf{a}^{\text{obs}}$ as well as possible vertex attributes $\mathbf{X} = \mathbf{x} \in \mathbb{R}^n$.
- A number of authors have proposed solutions, e.g. Liben-Nowell & Kleinberg (2003), Popescu & Ungar (2003) etc.
- Sometimes the "missingness" corresponds to a temporal component.

Link Prediction III



Picture from Kolaczyk 2010.

Link Prediction IV

- Often missingness is due to **sampling**. Then we must model the sampling mechanism.
- Often it is assumed that edge variables are missing **at random** (e.g. Hoff (2007)).
- The **at random** assumption corresponds to that the condition of whether A_{ij} is observed depends only on whether other $A_{i'j'}$ are observed not on the variable A_{ij} itself.
- If it were the case that the probability that A_{ij} was missing depended on the value of A_{ij} then this is called **informative missingness**.
- Given a suitable model for $\mathbf{X}$ and $\left(\mathbf{A}^{\text{miss}}, \mathbf{A}^{\text{obs}}\right)$ we might seek

$$\Pr\left\{\mathbf{A}^{\text{miss}} \mid \mathbf{A}^{\text{obs}} = \mathbf{a}^{\text{obs}}, \mathbf{X} = \mathbf{x}\right\}.$$

- Often researchers seek A_{ij}^{miss} separately for computational tractability.

Link Prediction V

- Can the change in a social network be modelled using features intrinsic to the network itself?
- If we think the network is evolving then we are predicting the formation of links.
- This can be model-based, say using preferential attachment or other evolutionary models.

Scoring Methods I

- For each pair i and j whose edge status is not known (for each $\{i,j\} \in V^{(2)}_{\text{miss}}$) a score value $s(i,j)$ is computed.
- Normally a set of predicted edges are found by setting a threshold $s^* > 0$ and then keeping the edges for which $s(i,j) > s^*$.
- A number of scores have been proposed.
- Scores are generally chosen to determine certain structural characteristics of a network graph $G^{(\text{obs})} = \left(V^{(\text{obs})}, E^{(\text{obs})}\right)$ associated with $\mathbf{A}^{\text{obs}} = \mathbf{a}^{\text{obs}}$.
- A simple choice would be

$$s(i,j) = -\operatorname{dist}_{G^{\text{obs}}}(u,v).$$

- The negative sign in this equation is included so that larger values indicate that vertices have increased probability of sharing an edge.

Scoring Methods II

- Also, there are also a number of scores based on the 1 -neighbourhood of node i; $N(i,1)$. We also define $N^{(\text{obs})}(i,1)$
- The so-called neighbourhood score is

$$s(i,j) = \left| N^{(\text{obs})}(i,1) \cap N^{(\text{obs})}(j,1) \right|.$$

- This assumes that in the missing data edges are more likely if observed edges exist between the nodes.
- As n grows in size this would naturally grow and so a normalized version is the Jaccard coefficient (see also our metrics lecture):

$$s(i,j) = \frac{\left| N^{(\text{obs})}(i,1) \cap N^{(\text{obs})}(j,1) \right|}{\left| N^{(\text{obs})}(i,1) \cup N^{(\text{obs})}(j,1) \right|}.$$

- This also relies on the observed and missing data having the same characteristics.

Scoring Methods III

- A score relating to the Jaccard coeficient is the Liben–Nowell score

$$s(i,j) = \sum_{k \in N^{(\mathrm{obs})}(i,1) \cap N^{(\mathrm{obs})}(j,1)} \frac{1}{\log \left| N^{(\mathrm{obs})}(k,1) \right|}.$$

- This score weights heavily the common neighbours of i and j that are not themselves highly connected.
- A problem with those scores is that they are heavily local.
- The implicit assumption to this choice is that vertices in the observed graph are tied implies that they should be linked also in the missing data.

Scoring Methods IV

- Link prediction can be approached as binary classification.
- We take $\mathbf{A}^{\text{obs}} = \mathbf{a}^{\text{obs}}$ as training data.
- Our goal is to use the data $\mathbf{a}^{\text{obs}}$ and any vertex attributes $\mathbf{X}$ to construct a classifier.
- Then we use the classifier to predict entries in $\mathbf{A}^{\text{miss}}$.
- We attempt to "approximate" the rule that determines the presence of edges.

Logistic Regression I

- We shall explain what we chose as $\mathbf{z}$ momentarily.
- Our models will be of the form:

$$\log\left\{\frac{\mathbb{P}_{\boldsymbol{\beta}}(A_{ij}=1 \mid \mathbf{Z}_{ij}=\mathbf{z})}{\mathbb{P}_{\boldsymbol{\beta}}(A_{ij}=0 \mid \mathbf{Z}_{ij}=\mathbf{z})}\right\} = \boldsymbol{\beta}^T\mathbf{z},$$

- here $\mathbf{Z}_{ij}$ is a vector of explanatory variables indexed by the pair $\{i,j\}$.
- Estimates of $\boldsymbol{\beta}$ can be produced using maximum likelihood.

Logistic Regression II

- Potential edges ij are classified as present or absent according to estimated classification probabilities:

$$\mathbb{P}_{\hat{\beta}}(A_{ij}^{\text{miss}} = 1 \mid \mathbf{Z}_{ij} = \mathbf{z}) = \frac{\exp\left(\hat{\beta}^T \mathbf{z}\right)}{1 + \exp\left(\hat{\beta}^T \mathbf{z}\right)},$$

  and if this quantity exceeds a threshold 0.5 the edge is deemed present.
- The $\mathbf{Z}_{ij}$ are here vectors of functions

$$\mathbf{Z}_{ij} = \left(g_1\left(\mathbf{A}^{\text{obs}}_{(-ij)}, \mathbf{X}\right), \ldots g_K\left(\mathbf{A}^{\text{obs}}_{(-ij)}, \mathbf{X}\right)\right)^T,$$

  where $\mathbf{A}^{\text{obs}}_{(-ij)}$ is all elements of $\mathbf{A}^{\text{obs}}$ barring A_{ij}.
- We only use A_{ij} to train our classifier.
- The functions $g_1(\cdot)$ up to $g_K(\cdot)$ are selected to parameterize predictive information in $\mathbf{A}^{\text{obs}} = \mathbf{a}^{\text{obs}}$ and $\mathbf{X} = \mathbf{x}$.

Logistic Regression III

- Standard logistic regression framework would have A_{ij} as independent.
- We do not know the effect on classifier accuracy dependence has.
- The underlying missingness mechanism should affect the answer.
- We shall think about logistic regression with latent variables.

Logistic Regression IV

- Let $\mathbf{M}$ we an unknown random symmetric $n \times n$ matrix of the form

$$\mathbf{M} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}.$$

  In this representation $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is a random but orthogonal matrix, $\mathbf{\Lambda}$ is a $n \times n$ random diagonal matrix, and $\mathbf{E} = (\epsilon_{ij})$ is a matrix of iid noise variables.
- To preserve symmetry we take $\epsilon_{ij} = \epsilon_{jj}$.
- The matrix $\mathbf{M}$ conditionally takes the form:

$$M_{ij} \mid \mathbf{U}, \mathbf{\Lambda}, \mathbf{E} = \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{ij}.$$

- We augment $\mathbf{Z}_{ij}$ by the unobserved variable M_{ij} so we get

$$\log \left\{ \frac{\mathbb{P}_\beta(A_{ij} = 1 \mid \mathbf{Z}_{ij} = \mathbf{z}, M_{ij} = m)}{\mathbb{P}_\beta(A_{ij} = 0 \mid \mathbf{Z}_{ij} = \mathbf{z}, M_{ij} = m)} \right\} = \beta^T \mathbf{z} + m.$$

- With this model, if we are only given $\mathbf{Z}_{ij}$ then $\{A_{ij}\}$ are dependent.

Logistic Regression V

- The dependence is a consequence of the hierarchical nature of our model.
- We introduced $\mathbf{M}$ to capture effects that are not captured by $\mathbf{Z}_{ij}$.
- To produce statistical predictions we need to specify distributions for $\mathbf{U}, \mathbf{\Lambda}$ and $\mathbf{E}$.
- For a Bayesian analysis, priors for β need to be specified.
- To determine if there are edges we use elements in $\mathbf{A}^{\text{miss}}$ we compute

$$\mathbb{E}\left(\frac{\exp\{\beta^T \mathbf{Z}_{ij} + M_{ij}\}}{1 + \exp\{\beta^T \mathbf{Z}_{ij} + M_{ij}\}} \mid \mathbf{A}^{\text{obs}} = \mathbf{a}^{\text{obs}}, \mathbf{Z}_{ij} = \mathbf{z} \right),$$

and compare this value to a given threshold.

Biclustering I

- We have looked at $G = (V, E)$ as an undirected network.
- We shall now generalize this to having two types of nodes; V_1 and V_2, respectively. There are edges between $v_1 \in V_1$ and $v_2 \in V_2$ given by the edge set E.
- Thus the intrinsic statistical object is (V_1, V_2, E), with $|V_1| = n$ and $|V_2| = m$. We assume the data form has **undirected** edges.
- The edges are collected in a **data matrix** X.
- The common inference problem addressed in terms of the data matrix is usually one of clustering.

Biclustering II

- Examples of biclustering applications include:
- For online retail X_{ij} can then show if user i wants product j, and our task is to segment users and products into relevant subgroups. Who might want product j ?
- In bioinformativs, X_{ij} could correspond to the log activation level of gene j in patient i. Our task is then to determine groups of patients with similar genetic profiles, while at the same time finding groups of genes with similar activation levels.
- In medicine determining groups in such data bases has helped to identify associations between active ingredients and adverse medical reactions.

Biclustering III

- Standard clustering can be applied either to rows or columns of a symmetric matrix A. Unlike A, X is not symmetric.
- Biclustering, clusters both these dimensions simultaneously.
- Biclustering algorithms identify groups of variables that display similar activity patterns under a specific subset of the common features.
- We write X_{ij} for the response of variable i under condition j.
- We can represent this in two ways by its set of rows $R = \left\{x_1^{(r)}, \ldots, x_n^{(r)}\right\}$ and by its set of columns $C = \left\{x_1^{(c)}, \ldots, x_m^{(c)}\right\}$.
- We will write $X = (R, C)$.
- We will also write $I \subset R$ and $J \subset C$.
- We will write X_{IJ} for the submatrix that has rows I and columns J.

Biclustering IV

- We define a **cluster of rows** as a subset of rows that exhibit similar behaviour across the set of all columns.
- A row cluster $X_{IC} = (I, C)$ is a subset of rows. Here $I = \{i_1, \ldots, i_k\}$ is a subset of rows ($I \subset R$ and $k \leq n$).
- A cluster of rows (I, C) can thus be defined as a k by m submatrix of the matrix X.
- A **cluster of columns** (in contrast) is a subset of columns that exhibit similar behaviour across the set of all rows.
- A column cluster $X_{RJ} = (R, J)$ is a subset of columns defined over the set of all rows R, where $J = \{j_1, \ldots, j_s\}$ is a subset of columns ($J \subset C$ and $s \leq m$).
- A cluster of columns (R, J) can then be defined as an n by s submatrix of the matrix X.

Biclustering V

- In contrast a bicluster is a subset of rows that exhibit similar behaviour across a subset of columns, and vice versa.
- The bicluster X_{IJ} is thus a subset of rows and a subset of columns where $I = \{i_1, \ldots, i_k\}$ is a subset of rows ($I \subset R$ and $k \leq n$), and $J = \{j_1, \ldots, j_s\}$ is a subset of columns ($J \subset C$ and $s \leq m$).
- A **bicluster** (I, J) can thus be defined as a k by s submatrix of the matrix X.
- Given a data matrix X we want to identify a set of biclusters $B_k = (I_k, J_k)$ so that B_k is in some sense homogeneous.
- A data matrix can be viewed as a bipartite graph.
- Recall, a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, is said to be **bipartite** if its vertices can be partitioned into two sets L and U such that every edge in E has exactly one end in L and the other element in U. Furthermore $V = L \cup U$.