Statistical analysis of network data lecture 13

Sofia Olhede

December 20, 2024

1. Biclustering
2. Extensions of Exchangeability
3. Graphex
4. Edge Exchangeability

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 $V = L \cup U$.

Biclustering VI

1. Biclusters with constant values.
2. Biclusters with constant values on rows or columns.
3. Biclusters with coherent values.
4. Biclusters with coherent evolutions.

The first three types here analyse the observed values in X; the fourth tries to identify coherent behaviour. To numerically summarize the data matrix we define

$$\bar{x}_{i,J} = \frac{1}{|J|}\sum_{j\in J} x_{ij}, \quad \bar{x}_{I,j} = \frac{1}{|I|}\sum_{i\in I} x_{ij}$$

$$\bar{x}_{I,J} = \frac{1}{|I||J|}\sum_{i\in I, j\in J} x_{ij}.$$

Biclustering VII

- First we define a perfect constant bicluster as a submatrix (I, J), where all values are equal, for all $i \in I$ and $j \in J$:

$$x_{ij} = \mu.$$

- Sometimes such "ideal" biclusters can be found in some data matrices.
- The values X_{ij} found in what can be considered a constant bicluster are generally presented as $\eta_{ij} + \mu$, where η_{ij} is the noise associated with the real value of X_{ij}.
- Hartigan introduced block clustering to find these groups. This algorithm splits the original data matrix into a set of submatrices (biclusters) and uses the variance to evaluate the quality of each bicluster (I, J):

$$\text{Var}\{I, J\} = \sum_{i \in I, j \in J} \{x_{ij} - \bar{X}_{I,J}\}^2. \quad (1)$$

Biclustering VIII

- To avoid continuous partitioning, Hartigan assumes that there are K biclusters within the data matrix.
- The algorithm stops when the data matrix is partitioned into K biclusters and the quality of the resulting biclustering is computed:

$$\text{Var}\{I, J\}_K = \sum_{k=1}^{K} \sum_{i \in I_k, j \in J_k} \{x_{ij} - \bar{X}_{I_k, J_k}\}^2. \tag{2}$$

- Hartigan designed a permutation-based method to induce the optimal number of biclusters, K.

Biclustering IX

- A perfect bicluster with constant columns is a submatrix (I, J), where all the values within the bicluster can be obtained using one of the following expressions:

$$\mathbb{E}\, x_{ij} = \mu + \alpha_i \tag{1}$$

$$\mathbb{E}\, x_{ij} = \mu \times \alpha_i. \tag{2}$$

  Here μ is the typical value within the bicluster and α_i is the adjustment for row $i \in I$.
- A perfect bicluster with constant rows is a submatrix (I, J), where all the values within the bicluster can be obtained using one of the following expressions:

$$\mathbb{E}\, x_{ij} = \mu + \beta_j \tag{3}$$

$$\mathbb{E}\, x_{ij} = \mu \times \beta_j. \tag{4}$$

  Here μ is the typical value within the bicluster and β_j is the adjustment for column $j \in J$.

- We have noticed that permutation invariance is a natural probabilistic symmetry for networks.
- We can either consider finite exchangeability, or a finite sample from an infinite exchangeable array.
- Exchangeability has a number of drawback:

  (a) Sparsity: The simplest (vertex) exchangeable representation is not consistent with sparsity;
  (b) Internal Heterogeneity: Neither are simple (vertex) exchangeable models easily brought together with "power-law degrees"
  (c) External Heterogeneity: It is more challenging to bring exchangeability together with covariates and other information.

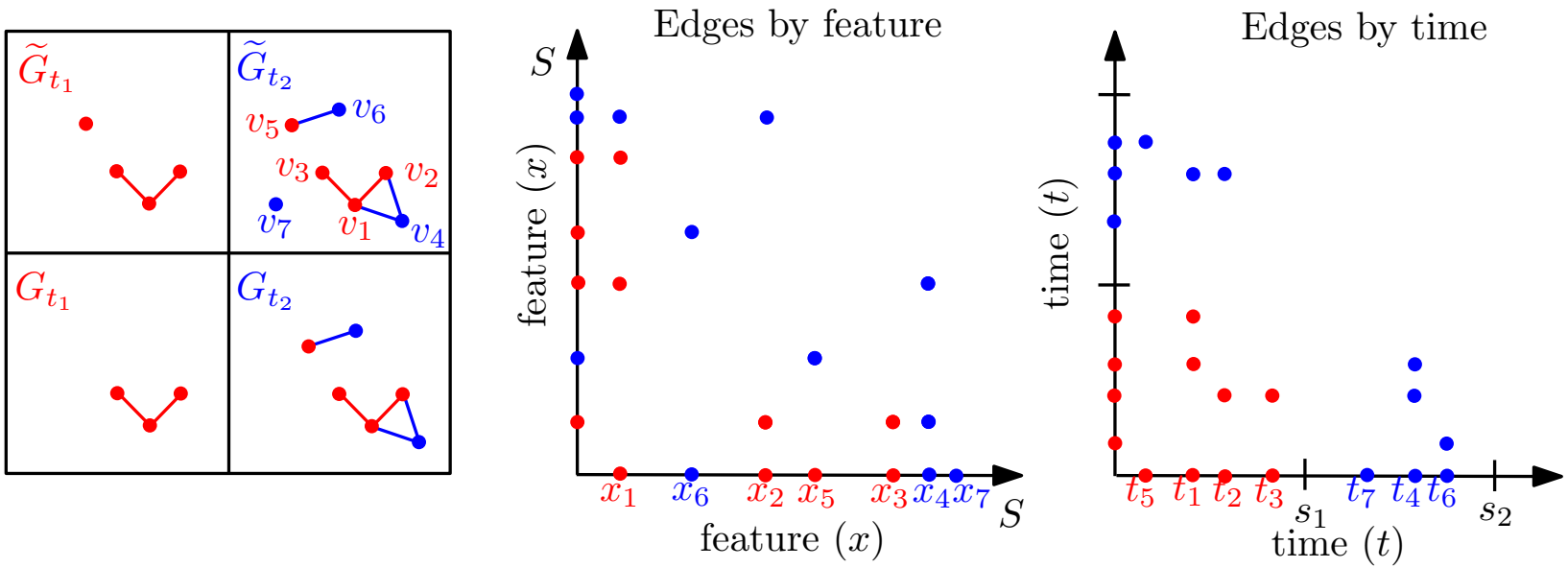
- What alternatives are there?

- The graphon Aldous–Hoover framework used a function $f(x, y), [0, 1]^2 \mapsto [0, 1]$ to parameterise the distrubtion of $\{A_{ij}\}_{i>j}$.
- As an alternative a network can be constructed from a point process.
- We define a point process $Y_t \subset [0, t] \times [0, t]$.
- We say Y_t is exchangeable if its distribution is unchanged by applying any measure preserving transformation.
- We observe A where A_{ij} is unity if $(\theta_i, \theta_j) \in Y$, where $\theta_1, \theta_2, \ldots$ are the arrival times of vertices in the point process.
- $\{Y_t\}$ being stationary does not mean X is exchangeable.
- The occurence of $(t, t') \in y$ means that there is an edge between vertices labelled t and t'.
- It is tempting to interpret t as the time a vertex enters the system; this is overinterpreting. But its connection to real data is easier to make with this interpretation.

- So then what? How do we get sparse and powerlaw?
- A graphex is a triplet (I, S, W). $I \in [0, \infty)$ (non-negative real number), $S: [0, \infty) \mapsto [0, \infty)$ is a measurable map, and $W: [0, \infty)^2 \mapsto [0, 1]^2$, a graphon function.
- We take realisations of a unit-rate point processes $\Xi = \{(\theta_i, \theta_j)\}$, $\Xi_i' = \{(\sigma_{ij}, \chi_{ij})\}$ and $\Xi'' = \{(\rho_j, \rho_j', \eta_j\}_j$. θ_i, σ_{ij} as potential vertex labels, while can be regarded as types of the corresponding labels.
- The point process determines the stochastic properties of θ_i.
- We then realize a graph A by a suitable combination of the three random arrays.
- The most important edges are present with probability/conditional expectation

$$\Pr\{W(\theta_i, \theta_j) \leq \zeta_{ij}\},$$

where $\{\zeta_{ij}\}$ are iid uniform random variables. Additional edges are present due to i) isolated 'stars' and ii) isolated 'edges'.



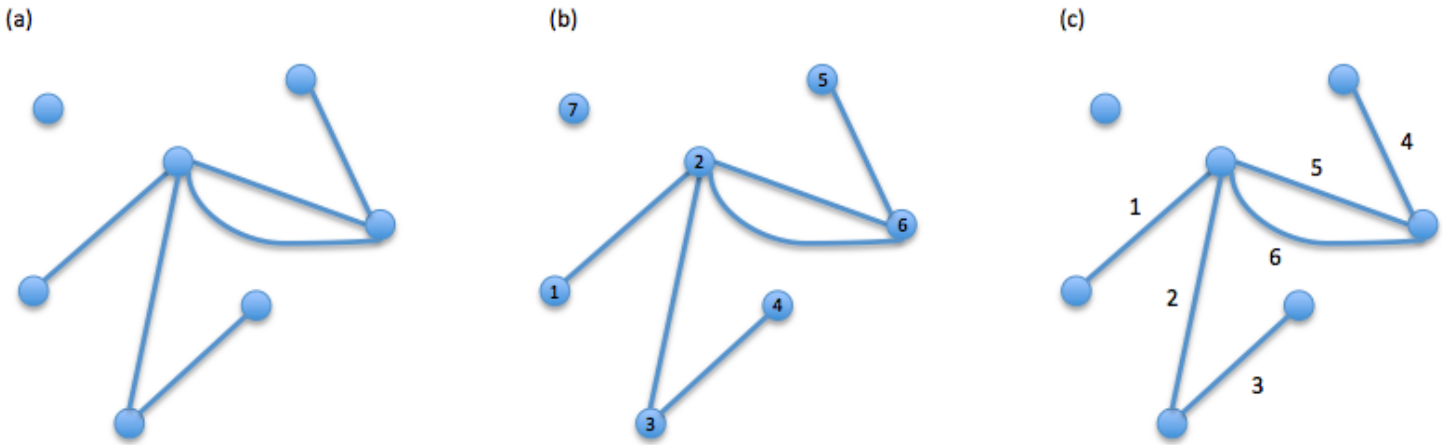
From Borgs, Chayes, Cohn and Holden 2018.

Edge Exchangeable RVs

- OK, but what if we start to observe edges rather that nodes?
- We assume we observe an edge list $E_n = \{E_i\}_{i\in[n]}$. Also assume we have permutations $\sigma : [n] \mapsto [n]$ and define $E_n^\sigma = \{E_{\sigma(i)}\}_{i\in[n]}$.
- We write a random edge labeled network $Y = \{Y_i\}_{i\in\mathbb{N}}$.
- We then have the following definition:

Definition (Edge exchangeability)

A randomly edge labeled network $\mathcal{Y} = \{\mathcal{Y}_i\}_{i\in\mathbb{N}}$ is edge exchangeable if $\mathcal{Y}^\sigma \stackrel{\mathcal{L}}{=} \mathcal{Y}$ for all permutations $\sigma : \mathbb{N} \to \mathbb{N}$.



From Crane and Dempsey 2018 (JASA).

- Edge exchangeable models have the same probability assigned to all edge labeled graphs that are equivalent up to a choice of relabeling.
- Any edge labelled network $\mathcal{Y} = \{\mathcal{Y}_i\}_{i\in\mathbb{N}}$ yields a compatible sequence of finite networks $\mathcal{Y} = \{\mathcal{Y}_i\}_{i\in\mathbb{N}}$ by taking $\mathcal{Y}_n = \mathcal{Y}|_n$ to be the restriction of subsampling $[n] \subset \mathbb{N}$. Any such sequence is infinitely edge exchangeable namely $\mathcal{Y}^\sigma \stackrel{\mathcal{L}}{=} \mathcal{Y}$ for all permutation $\sigma\ [n] \to [n]$ and $\mathcal{Y}_n|_{[m]} \stackrel{\mathcal{L}}{=} \mathcal{Y}_m$ for all $n \geq m \geq 1$.
- Even if all edges arrive in an exchangeable process, the vertices arrive in biased order weighted by the relative frequency of their occurrence in the network interactions.
- The sample of vertices, therefore, can be argued does not represent an exchangeable draw from the population of vertices.