The Deep Latent Position Block Model for Clustering and Representation of Networks

Rémi Boutin[1], Pierre Latouche[2] and Charles Bouveyron[3]

[1] LPSM - Sorbonne Université
[2] LMBP - Université Clermont Auvergne
[3] Maasai team - INRIA, Université Côte d'Azur

CEREMADE seminar, Université Paris Dauphine, 30th September 2024

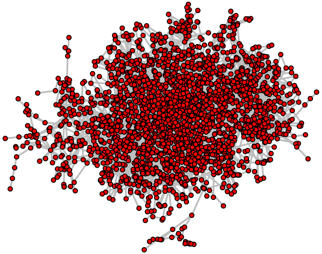
Outline

1. Introduction and motivation
2. Generative model
3. Inference
4. Evaluation on synthetic data
5. Analysis of a real world dataset
6. Conclusion

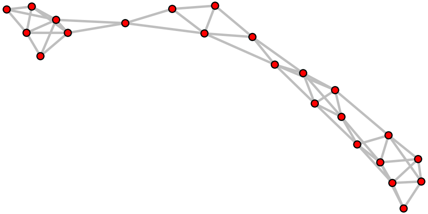
Introduction and motivation

The **networks** are a natural data structure to represent interactions between objects or individuals, such as:

- emails, co-authorship networks
- biological networks (protein-protein interactions networks)
- social websites (Facebook, Twitter)



(a) Cora network.



(b) Example of an enzyme network.

Figure: These networks representations were computed with the Fruchterman-Reingold algorithm[1].

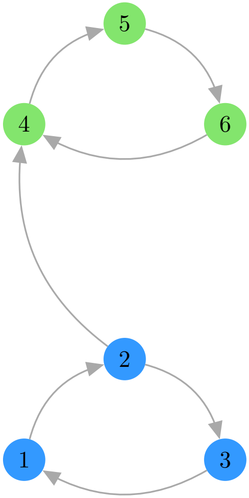
[1] Fruchterman, Reingold (1991).

Notations

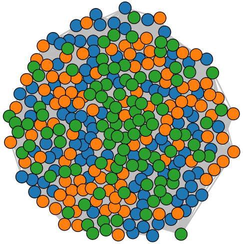
- i and j will refer to **nodes**.
- Adjacency matrix $\mathbf{A} \in \mathcal{M}_{N \times N}([0,1])$:

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected,} \\ 0 & \text{otherwise.} \end{cases}$$

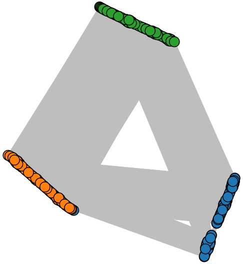
- q, k and r will refer to **clusters**.
- Q: the **number of clusters**.
- N: the **number of nodes**.
- M: the **number of edges**.
- $\mathrm{softmax}(x) = (1 + \sum_{k=1}^{K-1} e^{x_k})^{-1}(e^{x_1}, \ldots, e^{x_{K-1}}, 1)$, $\forall x \in \mathbb{R}^{K-1}$.



Introduction and motivation



(a) Fruchterman-Reingold node layout.



(b) Deep LPBM node layout.

Figure: Visualisations of **the same disassortative network** with two different node layouts. The node colour corresponds to their corresponding cluster. Two nodes within the same cluster (different clusters, respectively) connect with a probability of 0.01 (0.3).

Introduction and motivation

First line of work to obtain network visualisation is based on Physics and spring modelling[2]:

- Eades[3]
- Kamada and Kawai[4]
- Fruchterman-Reingold[5]
- Force Atlas 2 algorithm[6]

To summarise, nodes repulse from one another while edges attract.

[2] Hooke (1678).
[3] Eades (1984).
[4] Kamada, Kawai, et al. (1989).
[5] Fruchterman, Reingold (1991).
[6] Jacomy et al. (2014).

Introduction and motivation

The second line of work, from computational statistics and machine learning, estimates continuous latent representations of the nodes, and project them into a 2-dimensional space (or fix the dimension of the latent space to 2):

- latent position model (LPM)[7]
- latent position cluster model (LPCM)[8]
- autoencoders for graphs[9]

⚠ These methods *are not* compatible with block model approaches

[7] P. D. Hoff et al. (2002).
[8] Handcock et al. (2007).
[9] Kipf, Welling (2016).

Latent position models

The latent position model[10], as well as its extensions (including LPCM[11]) and many variational graph auto encoders[12] consider:

$$P(\mathbf{A}_{ij} = 1 \mid \eta_i, \eta_j) = \frac{1}{1 + e^{f(\eta_i, \eta_j)}}, \tag{1}$$

with

$$f(\eta_i, \eta_j) = \kappa - \|\eta_i - \eta_j\| \quad \text{or} \quad f(\eta_i, \eta_j) = \eta_i^\top \eta_j,$$

where $\kappa \in \mathbb{R}$, $\eta_i \in \mathbb{R}^p$ the latent node positions **to be used for visualisations**.

- They respect the transitivity property: *"the friend of my friend is my friend"* effect !
- They cannot handle disassortative graphs (such as star patterns).

[10] P. D. Hoff et al. (2002).
[11] Handcock et al. (2007).
[12] Kipf, Welling (2016).

Introduction and motivation

Few attempts to overcome this major drawback:

- Eigenvalue model[13]:

$$\mathbb{P}(\mathbf{A}_{ij} = 1 \mid \eta_i, \eta_j, \mathbf{\Pi}) = \Phi(\kappa + \boldsymbol{\eta}_i^\top \mathbf{\Pi} \boldsymbol{\eta}_j),$$

  with Φ the c.d.f of the normal distribution, $\mathbf{\Pi} \in \mathbb{R}^{Q \times Q}$ a diagonal matrix, $\kappa \in \mathbb{R}$ and $\boldsymbol{\eta}_i \in \mathbb{R}^Q$ the node latent representation.
- Extremal vertices model for random graph[14]

$$\mathbb{P}(\mathbf{A}_{ij} = 1 \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi}) = \boldsymbol{\eta}_i^\top \mathbf{\Pi} \boldsymbol{\eta}_j = \sum_{q,r=1}^{Q} \eta_{iq} \eta_{jr} \mathbf{\Pi}_{qr},$$

  where $\boldsymbol{\eta}_i \in \Delta_Q$ the Q-dimensional simplex, $\mathbf{\Pi} \in [0,1]^{Q \times Q}$.
- Generalised random dot product graph model[15]:

$$\mathbb{P}(\mathbf{A}_{ij} = 1 \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j) = \boldsymbol{\eta}_i^\top \mathbf{I}_{p,r} \boldsymbol{\eta}_j = \sum_{q=1}^{p} \eta_{iq} \eta_{jq} - \sum_{q=p+1}^{r+p} \eta_{iq} \eta_{jq},$$

  where $\boldsymbol{\eta}_i \in \mathcal{X}$ such that for any $x, y \in \mathcal{X}$, $x^\top \mathbf{I}_{p,q} y \in [0,1]$.

[13] P. Hoff (2007).
[14] Jean-Jacques Daudin et al. (2010).

Stochastic Block Model

The stochastic block model[16] assumes that each node is assigned to a single cluster:

$$\boldsymbol{\eta}_i \overset{i.i.d}{\sim} \mathrm{Multinomial}(1; \alpha = (\alpha_1, \ldots, \alpha_Q)). \tag{2}$$

where Q denotes the number of clusters. Hence,

$$\boldsymbol{\eta}_{iq} = \begin{cases} 1 & \text{if } i \text{ is in cluster } q, \\ 0 & \text{otherwise.} \end{cases}$$

Given the node cluster memberships, the probability of connection is given by:

$$\mathbf{A}_{ij} \mid \{\boldsymbol{\eta}_{iq} = 1, \boldsymbol{\eta}_{jr} = 1, \boldsymbol{\Pi}\} \sim \mathcal{B}(\boldsymbol{\Pi}_{qr}).$$

[16] Holland et al. (1983); Nowicki, Snijders (2001); Daudin et al. (2008).
[17] P. Hoff (2007); Jean-Jacques Daudin et al. (2010).

Stochastic Block Model

The stochastic block model[16] assumes that each node is assigned to a single cluster:

$$\boldsymbol{\eta}_i \overset{i.i.d}{\sim} \text{Multinomial}(1; \alpha = (\alpha_1, \ldots, \alpha_Q)). \tag{2}$$

where Q denotes the number of clusters. Hence,

$$\boldsymbol{\eta}_{iq} = \begin{cases} 1 & \text{if } i \text{ is in cluster } q, \\ 0 & \text{otherwise.} \end{cases}$$

Given the node cluster memberships, the probability of connection is given by:

$$\mathbf{A}_{ij} \mid \{\boldsymbol{\eta}_{iq} = 1, \boldsymbol{\eta}_{jr} = 1, \boldsymbol{\Pi}\} \sim \mathcal{B}(\boldsymbol{\Pi}_{qr}) = \mathcal{B}(\eta_i^\top \boldsymbol{\Pi} \eta_j).$$

Can we relax the binary constraint from $\boldsymbol{\eta}_i \in \{0,1\}^Q$ to $\boldsymbol{\eta}_i \in \Delta_Q$ instead ?[17]

[16] Holland et al. (1983); Nowicki, Snijders (2001); Daudin et al. (2008).
[17] P. Hoff (2007); Jean-Jacques Daudin et al. (2010).

Generative model

In this work, we assume that the node cluster membership assignment are not binary but continuous leading to the following assumptions:

$$\left.\begin{aligned} \mathbf{z}_i &\overset{i.i.d}{\sim} \mathcal{N}_{Q-1}(0, \mathbf{I}_{Q-1}) \\ \boldsymbol{\eta}_i &= \mathrm{softmax}(\mathbf{z}_i) \end{aligned}\right\} \quad \text{LogisticNormal distribution}$$

Generative model

In this work, we assume that the node cluster membership assignment are not binary but continuous leading to the following assumptions:

$$\left.\begin{array}{rcl} \mathbf{z}_i & \overset{i.i.d}{\sim} & \mathcal{N}_{Q-1}(0, \mathbf{I}_{Q-1}) \\ \boldsymbol{\eta}_i & = & \mathrm{softmax}(\mathbf{z}_i) \end{array}\right\} \text{ LogisticNormal distribution}$$

$$A_{ij} \mid \{\boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi}\} \sim \mathcal{B}(\boldsymbol{\eta}_i^\top \mathbf{\Pi} \boldsymbol{\eta}_j).$$

Link with other models

- Mixed-membership SBM[18]:
  - $U_{ij} \sim \mathrm{Multinomial}_Q(1; \eta_i)$ for the role of i
  - $U_{ji} \sim \mathrm{Multinomial}_Q(1; \eta_j)$ for the role of j
  - Marginalising over U_{ij}, U_{ji}, we retrieve the same probability as in Deep LPBM:

$$p(\mathbf{A}_{ij} = 1 \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi}) = \sum_{U_{ij}} \sum_{U_{ji}} p(\mathbf{A}_{ij} = 1 \mid \mathbf{U}_{ij}, \mathbf{U}_{ji}, \mathbf{\Pi}) p(\mathbf{U}_{ij} \mid \boldsymbol{\eta}_i) p(\boldsymbol{U}_{ji} \mid \boldsymbol{\eta}_j) = \boldsymbol{\eta}_i^\top \mathbf{\Pi} \boldsymbol{\eta}_j.$$

- Extremal vertices model for random graphs[19]:
  The quantity $p(\mathbf{A}_{ij} = 1 \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi})$ is similar to Deep LPBM. However:
  - $(\eta_i)_i$ are treated as parameters
  - the inference relies on a Taylor approximation of the likelihood preventing from using graph neural networks representational power.

[18] Airoldi et al. (2008).
[19] Jean-Jacques Daudin et al. (2010).

Link with other models

SBM considers **binary cluster memberships** $(\boldsymbol{\eta}_i)_i$, therefore, the conditional probability of connection between two nodes is:

$$p(\mathbf{A}_{ij} = 1 \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi}) = \sum_{q,r=1}^{Q} \eta_{iq}\eta_{jr}\mathbf{\Pi}_{qr} = \prod_{q,r=1}^{Q} \mathbf{\Pi}_{qr}^{\eta_{iq}\eta_{jr}}.$$

Let $p(\mathbf{A}_{ij} = 1 \mid \mathbf{\Pi}_{qr}) = \mathbf{\Pi}_{qr}$, the marginal probability of connection can be written using one of the following relaxations over $(\boldsymbol{\eta}_i)_i$:

- Canonical partial memberships[20]:

$$p(\mathbf{A}_{ij} = 1 \mid \mathbf{\Pi}) = \int_{\boldsymbol{\eta}_i, \boldsymbol{\eta}_j} \frac{1}{c} p(\boldsymbol{\eta}_i) p(\boldsymbol{\eta}_j) \prod_{q,r=1}^{Q} p\left(\mathbf{A}_{ij} = 1 \mid \mathbf{\Pi}_{qr}\right)^{\eta_{iq}\eta_{jr}} d\boldsymbol{\eta}_i d\boldsymbol{\eta}_j.$$

- Deep LPBM:

$$p(\mathbf{A}_{ij} = 1 \mid \mathbf{\Pi}) = \int_{\boldsymbol{\eta}_i, \boldsymbol{\eta}_j} p(\boldsymbol{\eta}_i) p(\boldsymbol{\eta}_j) \sum_{q,r=1}^{Q} \eta_{iq}\eta_{jr} p\left(\mathbf{A}_{ij} = 1 \mid \mathbf{\Pi}_{qr}\right) d\boldsymbol{\eta}_i d\boldsymbol{\eta}_j.$$

[20] Heller et al. (2008).

Inference

In this work, we aim at maximising the **marginal log-likelihood** given by:

$$\log p(\mathbf{A} \mid \mathbf{\Pi}) = \log \int_{\mathbf{Z}} p(\mathbf{A}, \mathbf{Z} \mid \mathbf{\Pi}) d\mathbf{Z}. \tag{3}$$

The marginal likelihood being intractable, we rely on a variational inference to maximise it. In particular, for any distribution $R(\cdot)$ over the latent variable $\mathbf{Z}$, the following decomposition holds true:

$$\log p(\mathbf{A} \mid \mathbf{\Pi}) = \mathscr{L}(\mathbf{\Pi}; R) + \mathrm{KL}(R(\cdot) \mid\mid p(\mathbf{Z} \mid \mathbf{A})),$$

where

$$\mathscr{L}(\mathbf{\Pi}; R) = \mathbb{E}_{R(\mathbf{Z})} \left[\log \frac{p(\mathbf{A}, \mathbf{Z} \mid \mathbf{\Pi})}{R(\mathbf{Z})} \right]. \tag{4}$$

The quantity $\mathscr{L}(\mathbf{\Pi}; R)$ is called the **expected lower bound (ELBO)**.

Inference: variational assumption

Assuming that the variational distribution respect the mean-field hypothesis:

$$R_\phi(\mathbf{Z}) \;=\; \prod_{i=1}^{N} \mathcal{N}_d(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i^2 \mathbf{I}_d), \tag{5}$$

where the parameters are the ouput of a graph convolutional network[21]:

$$(\mu_\phi(\mathbf{A}), \log \sigma_\phi(\mathbf{A})^2) = \mathrm{GCN}_\phi(\mathbf{A}). \tag{6}$$

[21] Kipf, Welling (2016).

GCN[23] and message passing

Denoting $h^0 = \mathbf{X}$ the node features, or $h^0 = \mathbf{I}_N$ if node features are not available, GCN is is message passing neural network[22]:

$$m_i^1 = \sum_{j \in \mathcal{N}(v)} \frac{\tilde{\mathbf{A}}_{ij}}{(\deg(i)\deg(j))^{\frac{1}{2}}} h_j^0, \qquad \text{message passing (=weighted average)}$$

$$h_i^1 = \mathrm{ReLu}\left((W^1)^\top m_i^1\right), \qquad \text{update of hidden state}$$

$$\mu_i = (W_\mu^2)^\top \sum_{j \in \mathcal{N}(v)} \frac{\tilde{\mathbf{A}}_{ij}}{(\deg(i)\deg(j))^{\frac{1}{2}}} h_j^1,$$

$$\log(\sigma_i^2) = (W_\sigma^2)^\top \sum_{j \in \mathcal{N}(v)} \frac{\tilde{\mathbf{A}}_{ij}}{(\deg(i)\deg(j))^{\frac{1}{2}}} h_j^1,$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$.

[22] Gilmer et al. (2017).
[23] Kipf, Welling (2016).

Details of the ELBO

Hence, the ELBO can be written as:

$$\begin{aligned}\mathscr{L}(\mathbf{\Pi}; R_\phi) = & \sum_{j<i} \mathbb{E}_{R_\phi(\mathbf{Z})}\left[\log p(\mathbf{A}_{ij} \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi})\right] - \sum_{i=1}^{N} \mathrm{KL}\left(R(\mathbf{z}_i) \mid p(\mathbf{z}_i)\right) \\ = & \sum_{j<i} \mathbf{A}_{ij} \mathbb{E}_{R_\phi(\mathbf{Z})}\left[\log(\boldsymbol{\eta}_i^\top \mathbf{\Pi} \boldsymbol{\eta}_j)\right] + (1 - \mathbf{A}_{ij}) \mathbb{E}_{R_\phi(\mathbf{Z})}\left[\log(1 - \boldsymbol{\eta}_i^\top \mathbf{\Pi} \boldsymbol{\eta}_j)\right] \\ & - \sum_{i=1}^{N} \frac{1}{2}\left(d\sigma_\phi(\mathbf{A})_i^2 + \|\mu_\phi(\mathbf{A})_i\|_2^2 - d \log \sigma_\phi(\mathbf{A})_i^2 - d\right),\end{aligned} \tag{7}$$

where $d = Q - 1$.

Next step: maximisation of $\mathscr{L}(\mathbf{\Pi}; R_\phi)$ with respect to $\mathbf{\Pi}$ and ϕ. We can directly optimise the previous quantity with respect to $\mathbf{\Pi}$ with gradient-based algorithm ... but not with respect to ϕ. Do you see the issue ?

The reparametrisation trick[24]

How to compute the gradient $\frac{\partial}{\partial \phi}\mathscr{L}(\mathbf{\Pi}; R_\phi)$? Based on the previous slide, we have:

$$\frac{\partial}{\partial \phi}\mathscr{L}(\mathbf{\Pi}; R_\phi) = \sum_{j<i} \frac{\partial}{\partial \phi} \mathbb{E}_{R_\phi(\mathbf{Z})}\left[\log p(A_{ij} \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi})\right] - \sum_{i=1}^{N} \frac{\partial}{\partial \phi} \overbrace{\mathrm{KL}\left(R_\phi(\mathbf{z}_i) \mid p(\mathbf{z}_i)\right)}^{\text{analytical form}}. \quad (8)$$

Issue: Since $R_\phi(\cdot)$ depends on ϕ, we cannot interchange the derivative and the integral in the term on the left-hand side.

[24] Kingma, Welling (2014); Rezende et al. (2014).

The reparametrisation trick[24]

How to compute the gradient $\frac{\partial}{\partial \phi}\mathscr{L}(\mathbf{\Pi}; R_\phi)$? Based on the previous slide, we have:

$$\frac{\partial}{\partial \phi}\mathscr{L}(\mathbf{\Pi}; R_\phi) = \sum_{j<i} \frac{\partial}{\partial \phi}\mathbb{E}_{R_\phi(\mathbf{Z})}\left[\log p(A_{ij} \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi})\right] - \sum_{i=1}^{N} \frac{\partial}{\partial \phi} \overbrace{\mathrm{KL}\left(R_\phi(\mathbf{z}_i) \mid p(\mathbf{z}_i)\right)}^{\text{analytical form}}. \tag{8}$$

Issue: Since $R_\phi(\cdot)$ depends on ϕ, we cannot interchange the derivative and the integral in the term on the left-hand side.

The **reparametrisation trick**[24] removes this dependency with the following sampling scheme:

$$\epsilon \sim \mathcal{N}_d(0, \mathbf{I}_d), \quad \text{and} \quad \mathbf{z}_i = \mu_\phi(\mathbf{A})_i + \sigma_\phi(\mathbf{A})_i \epsilon.$$

Hence, we can now interchange the integral and the derivative and use a Monte-Carlo estimate of the term on the right-hand side of the following equation:

$$\frac{\partial}{\partial \phi}\mathbb{E}_{R_\phi(\mathbf{Z})}\left[\log p(A_{ij} \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi})\right] = \frac{\partial}{\partial \phi}\mathbb{E}_{\epsilon}\left[\log p(A_{ij} \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi})\right] = \mathbb{E}_{\epsilon}\left[\frac{\partial}{\partial \phi}\log p(A_{ij} \mid \boldsymbol{\eta}_i, \boldsymbol{\eta}_j, \mathbf{\Pi})\right].$$

[24] Kingma, Welling (2014); Rezende et al. (2014).

Optimisation

Using the reparametrisation trick, we can now sample estimate of the gradients. Unfortunately, $\mathbf{\Pi}_{qr} \in]0,1[$, therefore, we use the following bijective mapping to get rid of the constraint:

$$f: \begin{cases} \mathbb{R} \longrightarrow]0,1[\\ x \longmapsto 0.5 + \pi^{-1} \arctan(x), \end{cases}$$

and its inverse

$$f^{-1}: \begin{cases}]0,1[\longrightarrow \mathbb{R} \\ \quad x \longmapsto \tan(\pi(x - 0.5 + \pi^{-1})). \end{cases}$$

Optimisation algorithm

Input: $C^{\textsf{KMeans}}$ labels provided by a KMeans on $\mathbf{A}$;
$\mathbf{Z}^0 = \mathrm{softmax}^{-1}(C^{\textsf{KMeans}})$;
for *epoch* $\in \{1, \dots, max\ iter_{init}\}$ **do**
  $\boldsymbol{\mu_\phi}, \boldsymbol{\sigma_\phi} \leftarrow \mathrm{Encoder}(\mathbf{A}; \phi)$;
  $\ell(\boldsymbol{\mu_\phi}, \boldsymbol{\sigma_\phi}, \mathbf{Z}^0) \leftarrow \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{\mu_{\phi,i}} - \boldsymbol{z}_i^0\|_2^2 + \|\sigma_{\phi,i}^2 - 0.01\|_2^2$;
  Stochastic gradient descent on $\ell(\boldsymbol{\mu_\phi}, \boldsymbol{\sigma_\phi}, \mathbf{Z}^0)$ with respect to ϕ;
end
for *epoch* $\in \{1, \dots,$ *max iter*$\}$ **do**
  $\boldsymbol{\mu_\phi}, \boldsymbol{\sigma_\phi} \leftarrow \mathrm{Encoder}(\mathbf{A}; \phi)$;
  $\mathbf{Z} \leftarrow \boldsymbol{\mu_\phi} \oplus (\boldsymbol{\sigma_\phi} \odot \epsilon)$;
  $\mathbf{\Pi} \leftarrow f(\tilde{\mathbf{\Pi}})$;
  $\hat{\boldsymbol{P}} \leftarrow \mathrm{Decoder}(\mathbf{Z}; \mathbf{\Pi})$;
  $\ell(\tilde{\mathbf{\Pi}}; \phi) \leftarrow$ Using $\hat{\boldsymbol{P}}, \mathbf{Z}, \boldsymbol{\mu_\phi}$ and $\boldsymbol{\sigma_\phi}$ in Equation (7);
  Stochastic gradient descent on $\ell(\tilde{\mathbf{\Pi}}; \phi)$ with respect to ϕ and $\tilde{\mathbf{\Pi}}$;
end

Theorem (Jean-Jacques Daudin et al. (2010))

Let $\boldsymbol{\eta} \in \mathcal{M}_{N\times Q}(\mathbb{R})$ *such that each row* $\boldsymbol{\eta}_i \in \Delta_Q$ *and* $\mathbf{\Pi} \in \mathcal{M}_{Q\times Q}([0,1])$*. Denoting* $\mathbf{P} \in \mathcal{M}_{N\times N}([0,1])$ *the matrix given by:*

$$\mathbf{P} = \boldsymbol{\eta}\mathbf{\Pi}\boldsymbol{\eta}^\top,$$

then, there exists $(\widetilde{\boldsymbol{\eta}}, \widetilde{\mathbf{\Pi}})$*, respecting the same conditions, such that* $(\widetilde{\boldsymbol{\eta}}, \widetilde{\mathbf{\Pi}}) \neq (\boldsymbol{\eta}, \mathbf{\Pi})$*, and:*

$$\widetilde{\mathbf{P}} = \widetilde{\boldsymbol{\eta}}\widetilde{\mathbf{\Pi}}\widetilde{\boldsymbol{\eta}}^\top = \boldsymbol{\eta}\mathbf{\Pi}\boldsymbol{\eta}^\top = \mathbf{P}.$$

Identifiability

In the following, we give sufficient conditions on a matrix $\mathbf{H}$ for $\widetilde{\boldsymbol{\eta}} = \boldsymbol{\eta}\mathbf{H}$ and $\widetilde{\mathbf{\Pi}} = \mathbf{H}^{-1}\mathbf{\Pi}(\mathbf{H}^\top)^{-1}$ to be correct candidates.

Lemma

Let $\mathbf{H} \in \mathcal{M}_{Q\times Q}(\mathbb{R})$ be a matrix such that:

1. $\mathbf{H}^{-1}$ *exists,*
2. $\mathbf{H}\mathbf{1}_{\boldsymbol{Q}} = \mathbf{1}_Q$, *where* $\mathbf{1}_Q = (1, \dots, 1)^\top$ *be the* Q*-dimensional vector made of* 1,
3. $\widetilde{\boldsymbol{\eta}} = \boldsymbol{\eta}\mathbf{H} \geq 0$,
4. $\mathbf{H}^{-1}\mathbf{\Pi}(\mathbf{H}^\top)^{-1} \in \mathcal{M}_{Q\times Q}([0,1])$.

Then:

(i) *For any* $i \in \{1, \dots, N\}$, $\widetilde{\boldsymbol{\eta}}_i^\top \mathbf{1}_{\mathbf{Q}} = \boldsymbol{\eta}_{\mathbf{i}}^\top \mathbf{H}\mathbf{1}_{\mathbf{Q}} = \boldsymbol{\eta}_{\mathbf{i}}\mathbf{1}_{\mathbf{Q}} = \mathbf{1}$, *i.e* $\boldsymbol{\eta}_i \in \Delta_Q$,

(ii) $\widetilde{\mathbf{\Pi}} \in \mathcal{M}_{Q\times Q}([0,1])$,

(iii) $\widetilde{\mathbf{P}} = \widetilde{\boldsymbol{\eta}}\widetilde{\mathbf{\Pi}}\widetilde{\boldsymbol{\eta}}^\top = \boldsymbol{\eta}\mathbf{H}\mathbf{H}^{-1}\mathbf{\Pi}(\mathbf{H}^\top)^{-1}\mathbf{H}^\top\boldsymbol{\eta}^\top = \boldsymbol{\eta}\mathbf{\Pi}\boldsymbol{\eta}^\top = \mathbf{P}$.

Model selection criteria

To select Q the number of clusters, we choose Akaike's information criterion (AIC)[25], the Bayesian information criterion (BIC)[26] as well as the integrated complete likelihood criterion (ICL)[27]:

$$\mathrm{AIC}(Q, \mathcal{M}) = \ln p(\mathbf{A} \mid \mathbf{Z}) - \frac{Q(Q+1)}{2} - N(Q-1),$$

$$\mathrm{BIC}(Q, \mathcal{M}) = \ln p(\mathbf{A} \mid \mathbf{Z}) - \frac{1}{2}\left(\frac{Q(Q+1)}{2} + N(Q-1)\right) \ln\left(\frac{N(N-1)}{2}\right),$$

$$\mathrm{ICL}(Q, \mathcal{M}) = \ln p(\mathbf{A} \mid \mathbf{Z}) - \frac{Q(Q+1)}{4} \ln\left(\frac{N(N-1)}{2}\right) + \ln p(\mathbf{Z}).$$

[25] Akaike (1974).
[26] Schwarz (1978).
[27] Biernacki et al. (2000).

Simulation setup

- Number of clusters = 5
- Number of nodes is set to 200
- β tunes for the level of connectivity between clusters
- $\epsilon = 0.01$ in all our experiments

$$\boldsymbol{\Pi}^\star = \begin{array}{ccc} \text{Communities} & \text{Disassortative} & \text{Hub} \\ \begin{pmatrix} \beta & \epsilon & \dots & \dots & \epsilon \\ \epsilon & \beta & \epsilon & \dots & \epsilon \\ \vdots & \epsilon & \beta & \dots & \epsilon \\ & \epsilon & \dots & \beta & \epsilon \\ \epsilon & \epsilon & \dots & \dots & \beta \end{pmatrix} & \begin{pmatrix} \epsilon & \beta & \dots & \dots & \beta \\ \beta & \epsilon & \beta & \dots & \beta \\ \beta & \beta & \epsilon & \dots & \beta \\ \beta & \beta & \dots & \epsilon & \beta \\ \beta & \beta & \dots & \dots & \epsilon \end{pmatrix} & \begin{pmatrix} \beta & \beta & \dots & \dots & \beta \\ \beta & \beta & \epsilon & \dots & \epsilon \\ \beta & \epsilon & \beta & \dots & \epsilon \\ \beta & \epsilon & \dots & \beta & \epsilon \\ \beta & \epsilon & \dots & \dots & \beta \end{pmatrix} \end{array}$$

Sampling strategies to evaluate the node partial memberships estimation

To evaluate the partial memberships estimation, we propose a new sampling scheme:

$$\eta_i^\star = \zeta \overline{\boldsymbol{\eta}}_i + (1-\zeta)\boldsymbol{\eta}_{unif} \in \Delta_Q,$$

where $\overline{\boldsymbol{\eta}}_i^\top = (0, \dots, 0, 1, 0, \dots)$, with a 1 on the q-th coordinate corresponding to the cluster of node i, $\boldsymbol{\eta}_{unif}^\top = (1/Q \cdots 1/Q) \in \Delta_Q$ the uniform probability vector and $\zeta \in (0,1)$ a parameter to tweak the level of noise.
Interpretation:

- The closer ζ is to 1 the closer the network is to a SBM sample.
- The closer ζ is to 0 the closer the network is to a Erdős–Rényi random graph.

Metric to evaluate the node partial memberships estimation

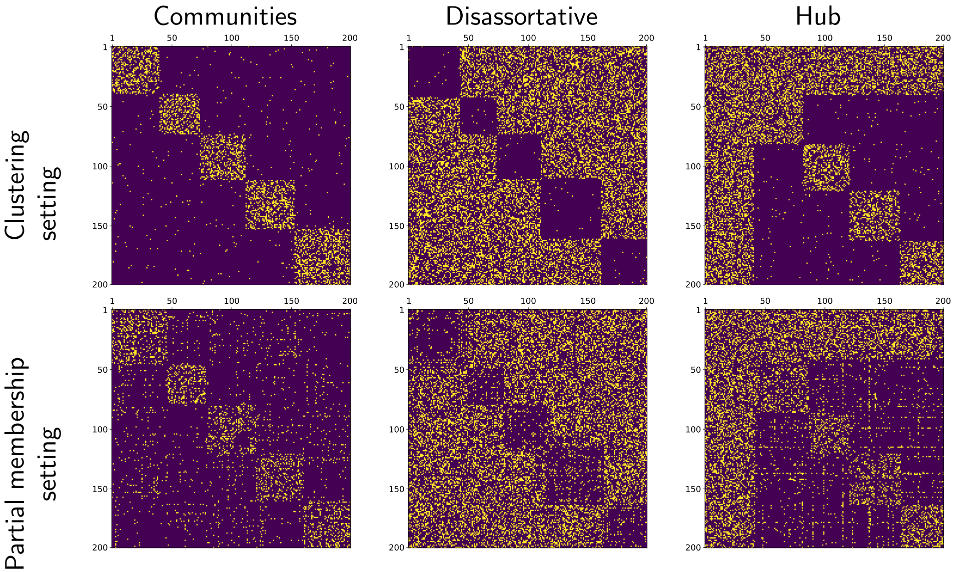
- **Metric for the partial memberships estimation**:
  To evaluate the relevance of $\hat{\boldsymbol{\eta}}$, we compare the amount of cluster membership shared between pairs of data points $\hat{\mathbf{U}} = \hat{\boldsymbol{\eta}}\hat{\boldsymbol{\eta}}^\top$ and the true ones $\mathbf{U}^\star = \boldsymbol{\eta}^\star\boldsymbol{\eta}^{\star\top}$. To do so, we compute the mean square-root of error[28]:

$$H = \sqrt{\frac{2}{N(N-1)} \sum_{i \leq j} |\mathbf{U}^\star_{ij} - \hat{\mathbf{U}}_{ij}|}. \tag{9}$$

- **Metric for the node clustering**: To evaluate the clustering results, we compare how close the obtained node partition and the true node partitions are by computing the **adjusted rand index (ARI)**. The closer it is to 1, the better.

[28] Heller et al. (2008); Latouche et al. (2014).

Adjacency matrices sampled according to our simulation schemes



Introductory example: the disassortative case

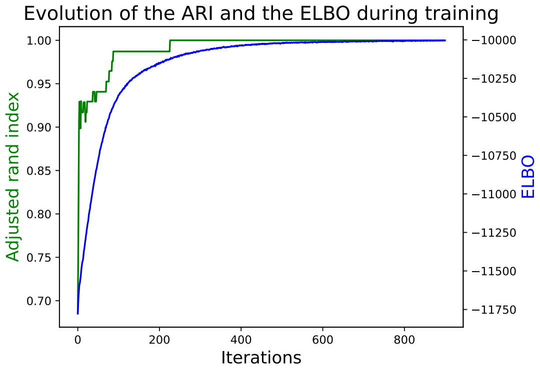
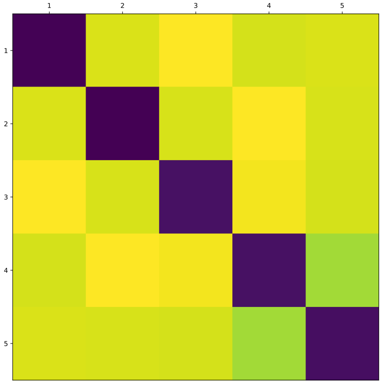
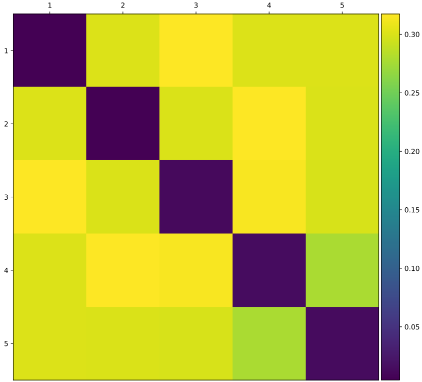


Figure: Evolution of the adjusted rand index and the ELBO during the estimation of Deep LPBM on a disassortative graph structure.

Introductory example: the disassortative case



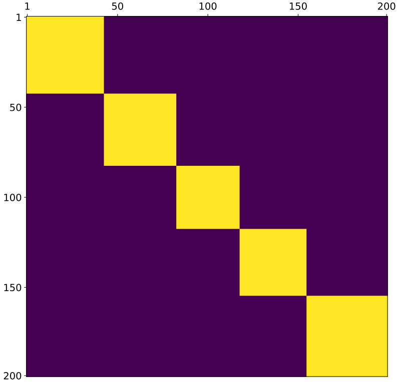
(a) True matrix $\mathbf{\Pi}$



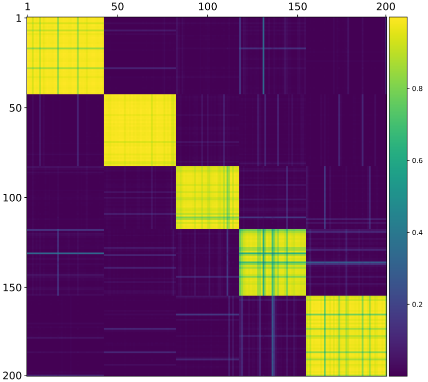
(b) Estimated matrix $\hat{\mathbf{\Pi}}$

Figure: On the left-hand side, the true connectivity matrix $\mathbf{\Pi}$ and on the right-hand side, the matrix estimated with Deep LPBM $\hat{\mathbf{\Pi}}$.

Introductory example: the disassortative case



(a) True matrix $\mathbf{U}^{\star}$.



(b) Estimated matrix $\hat{\mathbf{U}}$.

Figure: On the left-hand side, the true matrix $\mathbf{U}^{\star}$ computed from the one-hot encoded labels, on the right-hand side, the estimated matrix $\hat{\mathbf{U}}$ from $\hat{\boldsymbol{\eta}}$.

Partial memberships evaluation

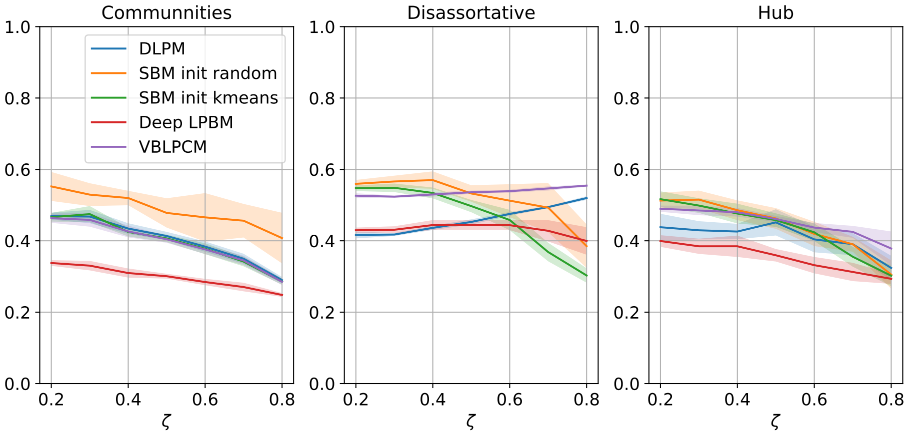


Figure: H-value for different values of ζ, **the lower, the better the estimation of η is.**

Benchmark: ARI on three different graph structures (the closer to 1 the better).

		Communities	Disassortative	Hub
$\beta = 0.2$	VBLPCM	0.98 ± 0.02	0.01 ± 0.00	0.72 ± 0.15
	DLPM	0.99 ± 0.01	0.00 ± 0.00	0.89 ± 0.10
	ARVGA	0.85 ± 0.03	0.01 ± 0.01	0.28 ± 0.06
	VGAE	0.97 ± 0.02	0.00 ± 0.01	0.64 ± 0.23
	SBM init kmeans	1.00 ± 0.01	1.00 ± 0.01	0.95 ± 0.10
	SBM init random	0.70 ± 0.03	0.45 ± 0.19	0.82 ± 0.16
	Deep LPBM	0.99 ± 0.01	0.39 ± 0.13	0.89 ± 0.09
$\beta = 0.3$	VBLPCM	1.00 ± 0.00	0.01 ± 0.01	0.79 ± 0.13
	DLPM	1.00 ± 0.00	0.00 ± 0.00	0.98 ± 0.01
	ARVGA	0.88 ± 0.03	0.06 ± 0.04	0.56 ± 0.22
	VGAE	1.00 ± 0.00	0.00 ± 0.01	0.72 ± 0.16
	SBM init kmeans	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	SBM init random	0.68 ± 0.15	0.79 ± 0.17	0.94 ± 0.13
	Deep LPBM	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01

Model Selection results

Deep LPBM most efficient model selection criterion is AIC, providing the following results:

Q	Commu	Disass	Hub
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	1
$5^\star$	**10**	**10**	**9**
6	0	0	0
10	0	0	0
16	0	0	0

Table: AIC's model selection for each network structure.

Real dataset: the French political blogosphere[29]

- This dataset is composed of 194 nodes
- Each node corresponds to a political blog
- An edge exists between two blogs if one of them possesses a hyperlink toward the other

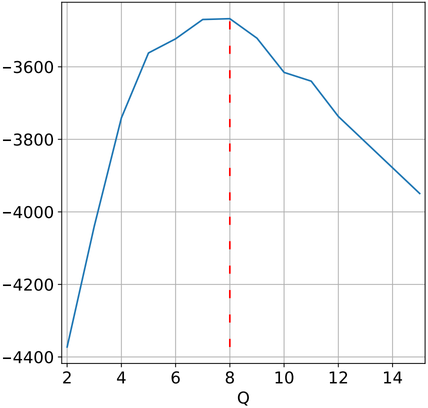
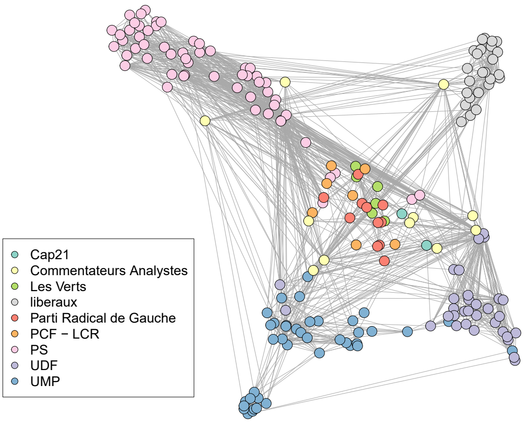
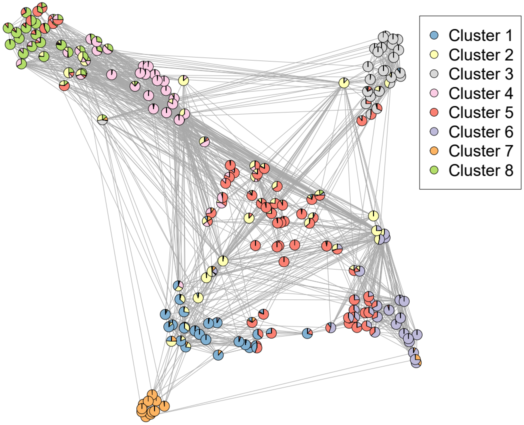


Figure: AIC values of Deep LPBM for Q varying from 2 to 15.

Real dataset: the French political blogosphere[30]



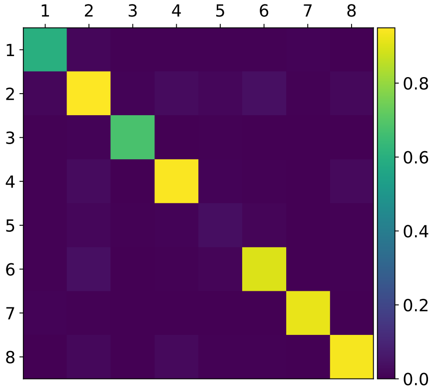
(a) Political parties.



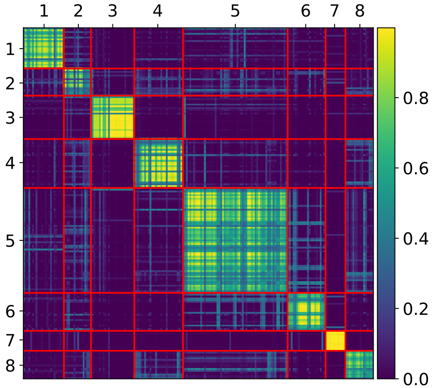
(b) Deep LPBM partial memberships.

Figure: Node positions estimated with Deep LPBM. On the right-hand side, the node colours indicate the political party associated to the blog. On the left-hand side, the estimated node partial memberships are represented by a pie chart.

Real dataset: the French political blogosphere[31]



(a) Estimated $\hat{\mathbf{\Pi}}$ for Q equal to 8.

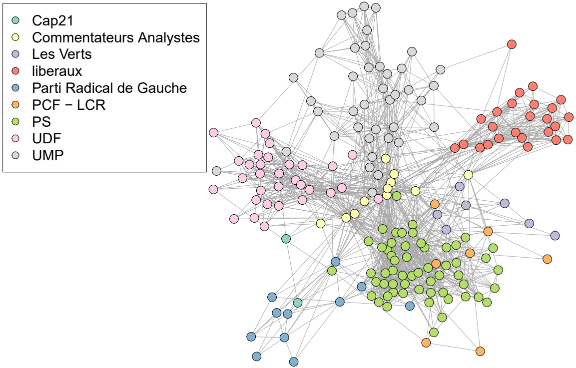


(b) Estimated $\hat{\mathbf{U}}$ defined in Section 4.

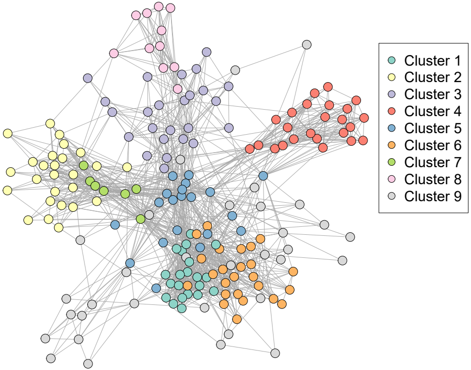
Figure: Visualisation of $\hat{\mathbf{\Pi}}$ and $\hat{\mathbf{U}}$ matrices. On the right-hand side, $\hat{\mathbf{U}}$ is a $N \times N$ matrix but is ordered by block which are delimited by the red lines.

[31] Zanghi et al. (2008).

Comparison with SBM results



(a) Political parties as node colours



(b) Clusters estimated by SBM.

Figure: The node positions were computed using a Fruchterman Reingold algorithm (Fruchterman, Reingold, 1991). On the left-hand side, the colour of the nodes corresponds to the political party the blog are associated with. On the right-hand side, the colour of the nodes indicate the SBM cluster assignments.

Conclusion

- The combination of graph neural networks with block modelling provides insightful results
- The model selection working without GNN still works in the variational autoencoder setting
- Need to test it on other datasets (in the presence of connectivity patterns different from communities)

Conclusion

- The combination of graph neural networks with block modelling provides insightful results
- The model selection working without GNN still works in the variational autoencoder setting
- Need to test it on other datasets (in the presence of connectivity patterns different from communities)

Thank you for your attention !

- Airoldi et al. (2008). "Mixed Membership Stochastic Blockmodels". In: *Journal of Machine Learning Research* 9(65), pp. 1981–2014.
- Akaike (1974). "A new look at the statistical model identification". In: *IEEE transactions on automatic control* 19(6), pp. 716–723.
- Biernacki, Celeux, Govaert (2000). "Assessing a mixture model for clustering with the integrated completed likelihood". In: *IEEE transactions on pattern analysis and machine intelligence* 22(7), pp. 719–725.
- Daudin, Picard, Robin (2008). "A mixture model for random graphs". In: *Statistics and computing* 18(2), pp. 173–183.
- Jean-Jacques Daudin, Pierre, Vacher (2010). "Model for heterogeneous random networks using continuous latent variables and an application to a tree–fungus network". In: *Biometrics* 66(4), pp. 1043–1051.
- Eades (1984). "A heuristic for graph drawing". In: *Congressus numerantium* 42(11), pp. 149–160.
- Fruchterman, Reingold (1991). "Graph drawing by force-directed placement". In: *Software: Practice and experience* 21(11), pp. 1129–1164.
- Gilmer et al. (2017). "Neural message passing for quantum chemistry". In: *International conference on machine learning*. PMLR, pp. 1263–1272.

Handcock, Raftery, Tantrum (2007). "Model-based clustering for social networks". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170(2), pp. 301–354.

Heller, Williamson, Ghahramani (2008). "Statistical models for partial membership". In: *Proceedings of the 25th International Conference on Machine learning*, pp. 392–399.

P. Hoff (2007). "Modeling homophily and stochastic equivalence in symmetric relational data". In: *Advances in neural information processing systems* 20, pp. 657–664.

P. D. Hoff, Raftery, Handcock (2002). "Latent space approaches to social network analysis". In: *Journal of the american Statistical association* 97(460), pp. 1090–1098.

Holland, Laskey, Leinhardt (1983). "Stochastic blockmodels: First steps". In: *Social networks* 5(2), pp. 109–137.

Hooke (1678). "De potentia restitutiva, or of spring explaining the power of springing bodies, vol. 1678". In: *London, UK: John Martyn* 23.

Jacomy et al. (2014). "ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software". In: *PloS one* 9(6), e98679.

Kamada, Kawai, et al. (1989). "An algorithm for drawing general undirected graphs". In: *Information processing letters* 31(1), pp. 7–15.

Kingma, Welling (2014). *Auto-Encoding Variational Bayes*. arXiv: `1312.6114 [stat.ML]`.

Kipf, Welling (2016). *Variational graph auto-encoders*. arXiv: `1611.07308 [stat.ML]`.

- Latouche, Birmelé, Ambroise (2014). "Model selection in overlapping stochastic block models". In: *Electronic Journal of Statistics* 8, pp. 762–794.
- Nowicki, Snijders (2001). "Estimation and prediction for stochastic blockstructures". In: *Journal of the American statistical association* 96(455), pp. 1077–1087.
- Rezende, Mohamed, Wierstra (2014). "Stochastic backpropagation and approximate inference in deep generative models". In: *International conference on machine learning*. Proceedings of Machine Learning Research, pp. 1278–1286.
- Rubin-Delanchy et al. (2022). "A statistical interpretation of spectral embedding: the generalised random dot product graph". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 84(4), pp. 1446–1473.
- Schwarz (1978). "Estimating the dimension of a model". In: *The annals of statistics*, pp. 461–464.
- Zanghi, Ambroise, Miele (Dec. 2008). "Fast online graph clustering via Erdős–Rényi mixture". In: *Pattern Recognition* 41, pp. 3592–3599.

Details about VGAE[32]

Denoting $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}(\mathbf{L} + \mathbf{I}_N)\mathbf{D}^{-1/2}$, the graph convolutional network can be summarised as

[32] Kipf, Welling (2016).

Details about VGAE[32]

Denoting $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}(\mathbf{L} + \mathbf{I}_N)\mathbf{D}^{-1/2}$, the graph convolutional network can be summarised as

$$\mu_\phi(\mathbf{A}) = \tilde{\mathbf{L}}\,\mathrm{ReLU}(\tilde{\mathbf{L}}\boldsymbol{\Omega}_0)\boldsymbol{\Omega}_\mu,$$
$$\log\sigma_\phi^2(\mathbf{A}) = \tilde{\mathbf{L}}\,\mathrm{ReLU}(\tilde{\mathbf{L}}\boldsymbol{\Omega}_0)\boldsymbol{\Omega}_\sigma,$$

where

- $\mathrm{ReLU}(x) = (\max(0, x_1), \ldots, \max(0, x_F))$ if $x \in \mathbb{R}^F$,
- $\boldsymbol{\Omega}_0 \in \mathcal{M}_{N\times D}(\mathbb{R})$ with $D = 64$ in all the experiments we carried out,
- $\boldsymbol{\Omega}_\mu, \boldsymbol{\Omega}_\sigma \in \mathcal{M}_{D\times(Q-1)}(\mathbb{R})$.

[32]Kipf, Welling (2016).

Model Selection

Table: Comparison of AIC (2a), BIC (2b) and ICL (2c) to select the best number of clusters for Deep LPBM with β equal to 0.3. The line corresponding to the true number of clusters, equal to 5, is highlighted and the most selected number of clusters is written in bold.

(a) AIC

Q	Commu	Disass	Hub
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	1
$5^{\star}$	**10**	**10**	**9**
6	0	0	0
10	0	0	0
16	0	0	0

(b) BIC

Commu	Disass	Hub
0	2	0
10	**8**	**10**
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

(c) ICL

Commu	Disass	Hub
0	2	0
10	**8**	**9**
0	0	1
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0