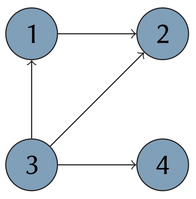
Automating the selection of preprocessing methods for deep neural networks

Marcus September, Francesco Sanna Passino
Department of Mathematics, Imperial College London
marcus.september22@imperial.ac.uk

Imperial College London

1. Latent position models and RDPGs

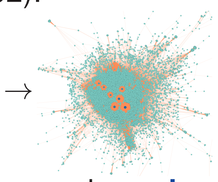
- Consider the **adjacency matrix** $\mathbf{A} = \{A_{ij}\} \in \{0,1\}^{n\times n}$ of a **graph**, where $A_{ij} = 1$ if node i connects to node j, and $A_{ij} = 0$ otherwise.



$$\Rightarrow \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Graph adjacency matrices can be modelled via **latent position models** (LPMs; Hoff et al., 2002):

$$\boldsymbol{x}_i \overset{iid}{\sim} F \;\rightarrow \mathbb{P}(A_{ij} = 1 | \boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \;\rightarrow$$



- LPMs are built on a powerful idea: expressing **edge-specific probabilities** through **latent node features** $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, using a **kernel function** $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow [0,1]$.
- If the **inner product** is used, the model is known as **random dot product graph** (RDPG; Athreya et al., 2018):

$$A_{ij} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \sim \text{Bernoulli}(\boldsymbol{x}_i^\intercal \boldsymbol{x}_j).$$

- RDPGs include many popular network models:
  - **Stochastic blockmodels** (SBMs): $\boldsymbol{x}_i = \boldsymbol{\mu}_{z_i}$ for a community $z_i \in \{1, \ldots, K\}$, giving a between-community connection probability $B_{k\ell} = \boldsymbol{\mu}_k^\intercal \boldsymbol{\mu}_\ell$;
  - **Degree-corrected SBMs**: $\boldsymbol{x}_i = \rho_i \boldsymbol{\mu}_{z_i}$ for $z_i \in \{1, \ldots, K\}$ and degree-correction $\rho_i \in (0,1)$.
- In RPDGs, the latent positions are **estimated via spectral decomposition** of the adjacency matrix.

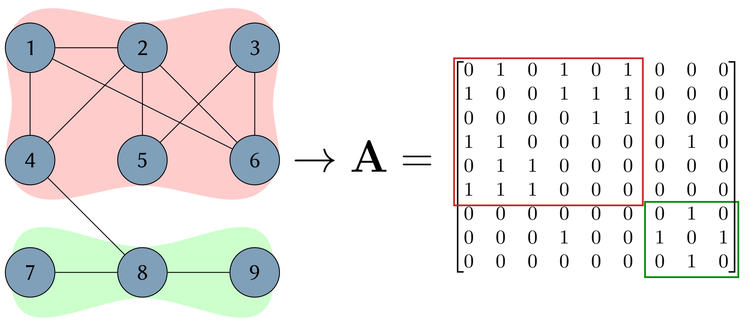
Definition 1: Adjacency spectral embedding (ASE)

For an integer $d \in \{1, \ldots, n\}$ and a binary *symmetric* adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$, the d-dimensional adjacency spectral embedding $\hat{\mathbf{X}} = [\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_n]^\intercal$ of $\mathbf{A}$ is

$$\hat{\mathbf{X}} = \boldsymbol{\Gamma} \boldsymbol{\Lambda}^{1/2} \in \mathbb{R}^{n\times d},$$

where $\boldsymbol{\Lambda}$ is a $d \times d$ diagonal matrix containing the absolute values of the d largest eigenvalues in magnitude, and $\boldsymbol{\Gamma}$ is a $n \times d$ matrix containing corresponding eigenvectors.

- For **directed** graphs, the **singular value decomposition (SVD)** is used.
- In practice, spectral embeddings often exhibit **manifold structure** (Rubin-Delanchy, 2020).
- **Spectral graph clustering** consists in unsupervised detection of groups of nodes from spectral embeddings.



$$\rightarrow \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- There are two **simultaneous challenges** in graph clustering via spectral embedding under the RDPG:

  Manifold structure
  Group structure
  } **Group-specific manifolds**.

- **Manifold** structure is accounted for by **latent structure models** (LSM, Athreya et al., 2021):
  - The latent positions $\boldsymbol{x}_i \overset{iid}{\sim} F$ are determined by draws from an underlying univariate distribution G on $[0,1]$, inducing F on a **univariate submanifold** $\mathcal{S} \subset \mathbb{R}^d$.
  - The distribution F on $\mathcal{S}$ is the distribution of the transformation $\boldsymbol{f}(\theta)$ of a univariate random variable $\theta \sim G$, where $\boldsymbol{f} : [0,1] \rightarrow \mathcal{S}$ is a function mapping θ to $\mathcal{S}$.
  - In simple terms, each node is assigned a draw θ_i from the underlying distribution G, representing *how far* along $\mathcal{S}$ the corresponding latent position lies:

$$\boldsymbol{x}_i = \boldsymbol{f}(\theta_i).$$

2. Proposed methodology: latent structure blockmodels (LSBMs)

- Add a **group structure** to LSMs $\rightarrow$ **latent structure blockmodels** (LSBMs).
- Each node is assigned a **latent membership** $z_i \in \{1, \ldots, K\}$, with probabilities $\eta_1, \ldots, \eta_K$, with $\eta_k \geq 0$ and $\sum_{k=1}^K \eta_k = 1$.
- Each community is associated with a **different one-dimensional structural support submanifold** $\mathcal{S}_k \subset \mathbb{R}^d$, $k = 1, \ldots, K$. Implicitly, $F = \sum_{k=1}^K \eta_k F_k$ is a mixture distribution with components $F_1, \ldots, F_K$ supported on $\mathcal{S}_1, \ldots, \mathcal{S}_K$.
- Assuming community allocations $\boldsymbol{z} = (z_1, \ldots, z_n)$, the latent positions are obtained as

$$\boldsymbol{x}_i | z_i \sim F_{z_i},\ i = 1, \ldots, n,$$

  where F_{z_i} is the distribution of the community-specific transformation $\boldsymbol{f}_{z_i}(\theta)$ of a shared univariate random variable $\theta \sim G$.
- G is common to *all* the nodes, and the pair (θ_i, z_i), where $\theta_i \sim G$, determines the latent position $\boldsymbol{x}_i$ through $\boldsymbol{f}_{z_i}$, such that:

$$\boldsymbol{x}_i = \boldsymbol{f}_{z_i}(\theta_i).$$

3. A Bayesian model for LSBMs

$$\begin{aligned}
\hat{\boldsymbol{x}}_i | \theta_i, \boldsymbol{f}_{z_i}, \boldsymbol{\sigma}^2_{z_i} &\sim \mathbb{N}_d \left\{ \boldsymbol{f}_{z_i}(\theta_i), \boldsymbol{\sigma}^2_{z_i} \mathbf{I}_d \right\},\ i = 1, \ldots, n, \\
\theta_i &\sim \mathbb{N}(\mu_\theta, \sigma^2_\theta),\ i = 1, \ldots, n, \\
f_{k,j} | \sigma^2_{k,j} &\sim \text{GP}(0, \sigma^2_{k,j} \xi_{k,j}),\ k = 1, \ldots, K,\ j = 1, \ldots, d, \\
\sigma^2_{k,j} &\sim \text{IG}(a_0, b_0),\ k = 1, \ldots, K,\ j = 1, \ldots, d, \\
z_i | \boldsymbol{\eta}, K &\sim \text{Categorical}(\boldsymbol{\eta}),\ i = 1, \ldots, n, \\
\boldsymbol{\eta} | K &\sim \text{Dirichlet}(\nu/K, \ldots, \nu/K), \\
K &\sim \text{Geometric}(\omega),
\end{aligned}$$

where $a_0, b_0, \nu, \omega, \sigma^2_\theta \in \mathbb{R}_+, \mu_\theta \in \mathbb{R}$, and $\xi_{k,j}$ is a positive semi-definite kernel function.

4. Posterior inference

- After marginalisation of $(f_{k,j}, \sigma^2_{k,j})$ and $\boldsymbol{\eta}$, inference is limited to the community allocations $\boldsymbol{z}$ and parameters $\boldsymbol{\theta}$.
- The posterior distribution $p(\boldsymbol{z}, \boldsymbol{\theta}, K \mid \hat{\mathbf{X}})$ is analytically intractable; inference is performed using **MCMC methods**.
  - Resample the community allocations $\boldsymbol{z}$;
  - Resample the latent parameters $\boldsymbol{\theta}$;
  - Split-merge communities;
  - Add or remove an empty community;
  - *If prior on kernels is used*: resample kernels.
- The kernel function is usually assumed to be in inner product form, with Zellner's g-prior on the scaling matrix.

5. Examples of LSBMs: SBMs, DCSBMs and quadratic LSBMs

- Kernels – SBM: $\xi_{k,j}(\theta, \theta') = \Delta_{k,j},\ \Delta_{k,j} \in \mathbb{R}_+$; DCSBM: $\xi_{k,j}(\theta, \theta') = \theta\theta' \Delta_{k,j}$; Quadratic: $\xi_{k,j}(\theta, \theta') = (\theta, \theta^2) \Delta_{k,j} (\theta', \theta'^2)^\intercal$.

(a): *SBM:* $f_{k,j}(\theta_i) = \nu_{k,j}$ **(b):** *DCSBM:* $f_{k,j}(\theta_i) = \theta_i \nu_{k,j}$ **(c):** *Quadratic:* $f_{k,j}(\theta_i) = \alpha_{k,j}\theta_i^2 + \beta_{k,j}\theta$

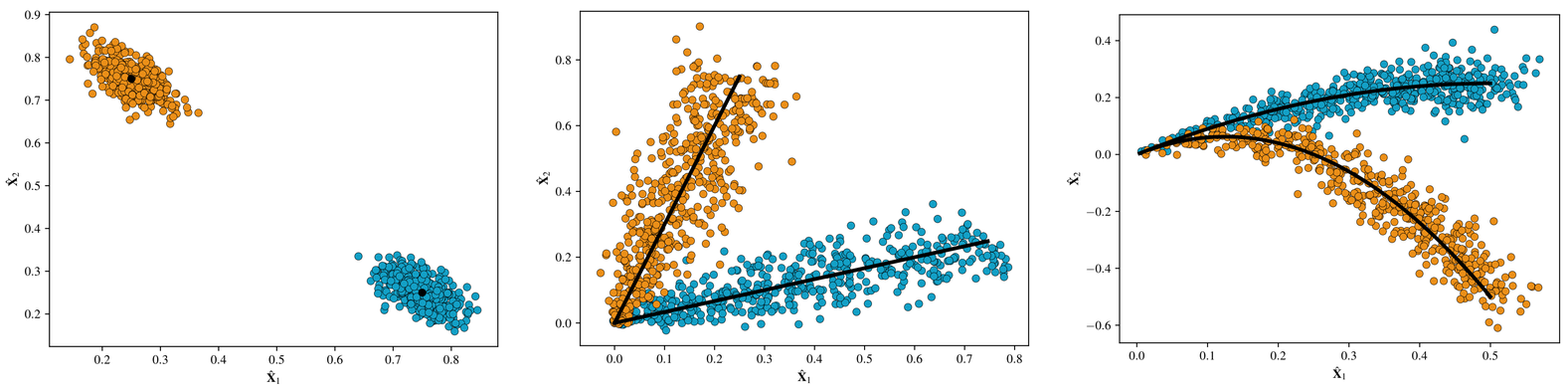


Figure 1: *Scatterplots of the two-dimensional ASE of simulated graphs arising from simple models, and true underlying latent curves (in black). For each graph, $n = 1000$ with $K = 2$ communities of equal size. For (a) and (b), $\boldsymbol{\nu}_1 = [3/4, 1/4]$, $\boldsymbol{\nu}_2 = [1/4, 3/4]$ and $\theta_i \sim$ Beta$(1,1)$. For (c), $\boldsymbol{\alpha}_k = [-1, -4]$, $\boldsymbol{\beta}_k = [1,1]$, $\boldsymbol{\gamma}_k = [0,0]$ and $\theta_i \sim$ Beta$(2,1)$.*

Method	LSBM($\hat{\mathbf{X}}$)	GMM($\hat{\mathbf{X}}$)	GMM($\tilde{\mathbf{X}}$)	SCSC($\hat{\mathbf{X}}$)	PGP($\hat{\mathbf{X}}$)	HLouvain	HClust($\hat{\mathbf{X}}$)
SBM	1.0	1.0	1.0	1.0	1.0	1.0	1.0
DCSBM	0.853	0.802	0.838	0.887	0.842	0.827	0.411
Quadratic LSBM	0.838	0.620	0.712	0.636	0.691	0.582	0.101

Table 1: *ARI for communities estimated using LSBM and alternative methodologies on the embeddings in Figure 1.*

6. Results on three networks

(a): *Harry Potter enmity graph* **(b):** *Drosophila connectome* **(c):** *ICL computer laboratories network*

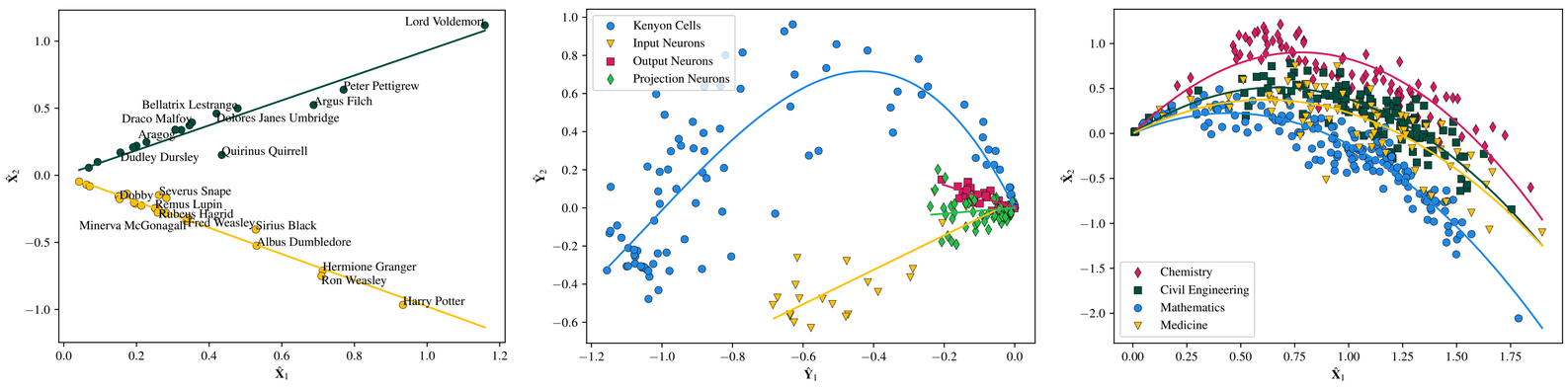


Figure 2: *Two-dimensional embeddings, estimated communities and true labels for three real-world dataset.*

Method	LSBM($\hat{\mathbf{X}}$)	GMM($\hat{\mathbf{X}}$)	GMM($\tilde{\mathbf{X}}$)	SCSC($\hat{\mathbf{X}}$)	PGP($\hat{\mathbf{X}}$)	HLouvain	HClust($\hat{\mathbf{X}}$)
Drosophila connectome	0.875	0.599	0.585	0.667	0.555	0.087	0.321
ICL computer laboratories	0.940	0.659	0.766	0.921	0.895	0.602	0.139

Table 2: *ARI for communities estimated using LSBM and alternative methodologies on the Drosophila and ICL laboratories networks.*

References

- Athreya, A. et al. (2018). "Statistical Inference on Random Dot Product Graphs: a Survey". Journal of Machine Learning Research 18, 1–92.
- Athreya, A. et al. (2021). "On Estimation and Inference in Latent Structure Random Graphs". Statistical Science 36.1, 68–88.
- Hoff, P. D. et al. (2002). "Latent space approaches to social network analysis". Journal of the American Statistical Association 97, 1090–1098.
- Rubin-Delanchy, P. (2020). "Manifold structure in graph embeddings". Advances in Neural Information Processing Systems 33, 11687–11699.

Paper and *python* library `lsbm`

Paper ***python*** **library** `lsbm`

Acknowledgements

This work is funded by the **Microsoft Security AI** research grant *"Understanding the enterprise: Host-based event prediction for automatic defence in cyber-security"*.