The Deep Latent Position Block Model

Rémi Boutin, Pierre Latouche, Charles Bouveyron

The Deep Latent Position Block Model

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

[1] *Université Paris Cité, CNRS, Laboratoire MAP5, UMR 8145, Paris, France*
[2] *Université Clermont Auvergne, CNRS, Laboratoire LMBP, UMR 6620, Aubière, France*
[3] *Université Côte d'Azur, CNRS, Laboratoire J.A.Dieudonné, INRIA, Maasai team, Nice, France*

Résumé.

L'augmentation des capacités de stockage et des données collectées a entraîné un accroissement des jeux de données disponibles. Par conséquent, l'utilisation des réseaux pour modéliser les relations entre différents objets, appelés des nœuds s'est accrue. Ces réseaux pouvant compter un très grand nombre de nœuds, l'information qu'ils contiennent doit-être résumée, le plus souvent à l'aide de méthodes de clustering de nœuds. Afin de rendre les résultats interprétables, une visualisation pertinente du réseau est également requise. Pour ce faire, nous proposons une nouvelle méthodologie appelée Deep-LPBM, permettant d'obtenir simultanément un clustering des noeuds basé sur une approche par bloc, plus générale que la détection de communautés, ainsi qu'une représentation continue des nœuds dans un espace latent. Deep-LPBM utilise une stratégie d'auto encodeur variationnel, s'appuyant sur un réseau de convolution de graphe, avec un décodeur adapté. L'inférence repose sur la vraisemblance marginale du modèle, et l'optimisation alterne entre des équations analytiques ainsi qu'une descente de gradient stochastique. Ce travail étant en cours, des expériences sur données simulées ainsi que sur données réelles seront fournies si le papier est accepté pour une présentation orale.

Mots-clés. Clustering de nœuds, auto-encodeur de graphe variationnel, modélisation par blocs, visualisation de graphe

Abstract. The increase in the quantity of data has led to a soaring use of networks to model relationships between different objects, called nodes. Since the number of nodes can be very large, the network information must be summarised, mostly with node clustering methods. In order to make the results interpretable, a relevant visualization of the network is also required. To tackle those two issues, we propose a new method called Deep-LPBM which provides simultaneously a network visualization based on block modelling, allowing a more general clustering than community detections, as well as a continuous representation of nodes in a latent space. Our methodology is based on a variational autoencoder strategy, relying on a graph convolutional network, with a specifically designed decoder. The inference is based on the marginal likelihood of the model, and the optimisation combines analytical equations with stochastic gradient descent. As this work is ongoing, experiments on simulated data as well as on real data will be provided if the paper is accepted for an oral presentation.

Keywords. Node clustering, variational graph auto-encoder, block modelling, network visualisation

†mail: remi.boutin.stat@gmail.com

1 Introduction and contribution

Networks are encountered in a variety of fields, ranging from social sciences to biology. Their capacity to represent any type of object and relationship makes it a core object to model interactions. However, they present difficulties to apprehend since non-observed features, such as node cluster memberships, may impact the observed network topology and engender specific connectivity patterns. This requires the development of specifically fashioned methods, models and inference strategies to capture information. To give an example, one of the most studied type of groups is called a community and corresponds to nodes highly connected to nodes of the same group but poorly connected to nodes from other groups. The community-detection methods are numerous but do not necessarily generalise to other types of structure such as a star pattern (a cluster of nodes poorly connected together but highly connected to nodes from the other clusters). Therefore, being able to capture the structure underlying the data is essential to model any type of relationship, even without prior knowledge of the network topology. This flexibility was provided by the stochastic block model (SBM, Snijders and Nowicki, 1997; Daudin, Picard, et al., 2008), enabling to model any type of connectivity patterns among the network. However, this flexibility comes at the cost of representation. Indeed, those methodologies do not provide a direct representation of the network, but only a high-level depiction of the underlying patterns, where clusters are represented as nodes, cluster sizes as node sizes, and the number of connections between clusters as edge widths. This meta representation may hide some node-specific properties. For instance, a node in a cluster may be more connected to another group than the other nodes in its cluster. Hence, this feature should appear in the network representation, since this node might play a crucial role in the network, precisely because it connects two clusters. To represent a network, positional approaches have been proposed by performing link prediction based on the similarity between estimated continuous node representations (Hoff et al., 2002; Handcock et al., 2007). Unfortunately, such methods only estimate communities. While some previous work has focused on combining the two (Hoff, 2007; Daudin, Pierre, et al., 2010), the methodology we propose is the first to incorporate the graph neural network ability to create informative node embeddings. To this end, a marginalised block model is introduced, where a logistic-Gaussian distribution models the node cluster membership probabilities and is used for visualisation purposes. This is an ongoing work that will come with a Python package as well as an extensive benchmark and a real world use-case.

Notations In this paper, matrices and collections of vectors are denoted in bold cases $\mathbf{X}$, the space of $n \times m$ matrices with coefficient in E is denoted $\mathcal{M}_{n\times m}(E)$, and should not be confused with the multinomial distribution denoted $\mathcal{M}_n(m, p)$ where n is the dimension of the vector, m is the number of draws and $p = (p_1, \dots, p_n) \in \Delta_n$ is the probability vector. The n-dimensional simplex is denoted $\Delta_n = \{p \in \mathbb{R}^n : \forall i, p_i \geq 0 \text{ and } \sum_{i=1}^n p_i = 1\}$. Moreover, the network is denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \dots, N\}$ denotes the set of vertices and $\mathcal{E}$ the set of edges, and the binary adjacency matrix $\mathbf{A} \in \mathcal{M}_{N\times N}(\{0, 1\})$ such that $A_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$. In this work, the graph is assumed to be directed, meaning that $\mathbf{A}$ does not need to be symmetric.

2 Assumptions regarding the network generation

In this section, we present the assumptions concerning the graph generation. Assuming that the number of clusters Q is fixed beforehand, each node $i \in \mathcal{V}$ is assumed to belong to a cluster, represented by the cluster membership vector C_i. The variables $(C_i)_i$ are assumed to be independent and identically distributed (i.i.d) according to a multinomial distribution such that:

$$C_i \overset{i.i.d}{\sim} \mathcal{M}_Q(1, \gamma),$$

with $\gamma \in \Delta_Q$ the vector of cluster proportions. The vectors $C_i \in \{0,1\}^Q$ are one-hot encoded, with $C_{iq} = 1$ if node i belongs to cluster q and $C_{iq} = 0$ otherwise. The probability of the cluster membership matrix $\mathbf{C} = (C_1, \dots, C_N)^T \in \mathcal{M}_{N\times Q}(\{0,1\})$ is given by:

$$p(\mathbf{C} \mid \gamma) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \gamma_q^{C_{iq}}. \tag{1}$$

Given the cluster membership matrix $\mathbf{C}$, the nodes are assumed to be independent, and represented by a Gaussian vector Z_i in a $Q-1$ dimensional latent space:

$$Z_i \mid C_{iq} = 1 \overset{i.i.d}{\sim} \mathcal{N}_{Q-1}\left(\mu_q, \sigma_q^2 \mathbf{I}_{Q-1}\right). \tag{2}$$

The set of node embeddings is denoted $\mathbf{Z} = (Z_i)_i$ in the rest of the paper, and the set of means and variances are denoted $\boldsymbol{\mu} = (\mu_q)_q$ and $\boldsymbol{\sigma}^2 = (\sigma_q^2)_q$ respectively. To link the latent representations of the nodes Z_i, with the block modelling, we rely on the *bijective softmax* transformation, as presented in Xu et al. (2014), $h\colon Z_i \in \mathbb{R}^{Q-1} \mapsto \eta_i \in \Delta_Q$ where:

$$\eta_{iq} = \begin{cases} e^{Z_{iq}} / \left(1 + \sum_{r=1}^{Q-1} e^{Z_{ir}}\right) & \text{if } q \in \{1, \dots, Q-1\} \\ 1 / \left(1 + \sum_{r=1}^{Q-1} e^{Z_{ir}}\right) & \text{if } q = Q \end{cases}, \tag{3}$$

and we denote $\boldsymbol{\eta} = (\eta_1, \dots, \eta_N)^\top \in \mathcal{M}_{N\times Q}\big((0,1)\big)$. The mapping h aims at encoding the $(Z_i)_i$ into cluster membership probabilities. Eventually, the probability of connection between two nodes follows a Bernoulli distribution with parameters depending on η such that:

$$p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi}) = \prod_{i\neq j} p(A_{ij} \mid Z_i, Z_j, \mathbf{\Pi}) = \prod_{i\neq j} \left(\eta_i^\top \mathbf{\Pi} \eta_j\right)^{A_{ij}} \left(1 - \eta_i^\top \mathbf{\Pi} \eta_j\right)^{1-A_{ij}}, \tag{4}$$

where $\mathbf{\Pi} = (\pi_{qr})_{1\leq q,r\leq Q} \in \mathcal{M}_{Q\times Q}\big((0,1)\big)$ is the matrix of probability of connection between clusters. Consequently, the joint distribution of $(\mathbf{A}, \mathbf{Z}, \mathbf{C})$ can be factorised as:

$$p(\mathbf{A}, \mathbf{Z}, \mathbf{C} \mid \mathbf{\Pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}) = p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi}) p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(\mathbf{C} \mid \boldsymbol{\gamma}). \tag{5}$$

3 Inference

The next section presents the inference as well as the optimisation.

3.1 Likelihood

To estimate the parameters, we rely on the marginal likelihood of the network, with latent variables $\mathbf{C}$ and $\mathbf{Z}$, and the set of parameters $\Theta = \{\mathbf{\Pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}\}$. From Equations (1), (2) and (4), we can deduce that the marginal log-likelihood is given by:

$$\mathcal{L}(\Theta; \mathbf{A}) = \log p(\mathbf{A} \mid \Theta) = \log \left(\sum_{\mathbf{C}} \int_{\mathbf{Z}} p(\mathbf{A}, \mathbf{C}, \mathbf{Z} \mid \Theta) d\mathbf{Z} \right). \tag{6}$$

Unfortunately, this quantity is not tractable since the sum over $\mathbf{C}$ requires to compute Q^N terms. Therefore, we choose to rely on a variational inference strategy for approximation purposes.

Decomposition of the marginal log-likelihood For any distribution $R(\mathbf{C}, \mathbf{Z})$, the following decomposition holds:

$$\mathcal{L}(\Theta; \mathbf{A}) = \mathscr{L}(R(\cdot); \Theta) + \mathrm{KL}\left(R(\cdot) || p(\mathbf{C}, \mathbf{Z} \mid \mathbf{A})\right), \tag{7}$$

where the expected lower bound (ELBO) is given by:

$$\mathscr{L}(R(\cdot); \Theta) = \mathbb{E}_R \left[\log \frac{p(\mathbf{A}, \mathbf{C}, \mathbf{Z} \mid \Theta)}{R(\mathbf{C}, \mathbf{Z})} \right]. \tag{8}$$

Since the Kullback-Leibler divergence is always positive in Equation (7), the ELBO is a lower bound of the marginal log-likelihood. Since the marginal log-likelihood does not depend on $R(\cdot)$, maximizing the ELBO with respect to $R(\cdot)$ is equivalent to minimizing the Kullback-Leibler divergence between $R(\cdot)$ and the posterior distribution. Hence, we restrict the family of variational distributions by considering a mean-field assumption as well as the following hypotheses to make the ELBO tractable:

$$R(\mathbf{C}, \mathbf{Z} \mid \mathbf{A}) = R(\mathbf{C}) R(\mathbf{Z} \mid \mathbf{A}), \tag{9}$$

$$R(\mathbf{C}) = \prod_{i=1}^{N} R_{\tau_i}(C_i) = \prod_{i=1}^{N} \mathcal{M}_Q(C_i; 1, \tau_i), \tag{10}$$

$$R(\mathbf{Z} \mid \mathbf{A}) = \prod_{i=1}^{N} R_\phi(Z_i \mid \mathbf{A}) = \prod_{i=1}^{N} \mathcal{N}_{Q-1}(Z_i; \mu_\phi(\mathbf{A})_i, \sigma^2_\phi(\mathbf{A})_i \mathbf{I}_{Q-1}), \tag{11}$$

where $\boldsymbol{\tau} = (\tau_i)_{i=1}^N$ with $\forall i \in \{1, \dots, N\}$, $\tau_i \in \Delta_Q$. Moreover, in Equation (11), the mapping $\mu_\phi : \mathcal{M}_{N \times N}(\mathbb{R}) \mapsto \mathcal{M}_{N \times (Q-1)}(\mathbb{R})$ ($\sigma^2_\phi : \mathcal{M}_{N \times N}(\mathbb{R}) \mapsto (\mathbb{R}^+)^N$ respectively) is the mapping normalising the adjacency matrix (with 1 on its diagonal for numeric stability) by its degree, $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I}_N)\mathbf{D}^{-1/2}$, and encoding the normalised adjacency matrix into the approximated posterior means (standard deviations) of the node latent positions. The diagonal matrix $\mathbf{D}$ is filled with $D_{ii} = \sum_{j=1}^{N} (\mathbf{A} + \mathbf{I}_N)_{ij}$. The two mappings μ_ϕ and σ^2_ϕ rely on a GCN parametrised by ϕ. Regarding the encoder of the adjacency matrix, we based our neural network architecture on Kipf and Welling (2016). Thus, the ELBO can be decomposed as follows:

$$
\begin{aligned}
\mathscr{L}(R(\cdot);\Theta) =& \mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi})\right] + \mathbb{E}_R\left[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})\right] - \mathbb{E}_R\left[\log R(\mathbf{Z} \mid \mathbf{A})\right] \\
&+ \mathbb{E}_R\left[\log p(\mathbf{C} \mid \boldsymbol{\gamma})\right] - \mathbb{E}_R\left[\log R(\mathbf{C})\right]. \\
=& \underbrace{\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi})\right]}_{\textbf{Reconstruction loss}} \underbrace{- \mathrm{KL}(R(\mathbf{Z} \mid \mathbf{A}) || p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})) - \mathrm{KL}(R(\mathbf{C}) || p(\mathbf{C} \mid \boldsymbol{\gamma}))}_{\textbf{Regularising term}} \\
=& \sum_{i,j=1}^{N} \Big(A_{ij} \mathbb{E}_R\left[\log \eta_i^\top \mathbf{\Pi} \eta_j\right] + (1 - A_{ij}) \mathbb{E}_R\left[\log\left(1 - \eta_i^\top \mathbf{\Pi} \eta_j\right)\right] \Big) \qquad (12) \\
&- \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \, \mathrm{KL}_{iq}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i, \mu_q, \sigma_q\right) - \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \frac{\tau_{iq}}{\gamma_q},
\end{aligned}
$$

where

$$
\mathrm{KL}_{iq}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i, \mu_q, \sigma_q\right) = \log \frac{\sigma_q^{(Q-1)}}{\sigma_\phi(\mathbf{A})_i^{(Q-1)}} - \frac{Q-1}{2} + \frac{(Q-1)\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2}{2\sigma_q^2}.
$$

3.2 Optimisation

To optimise the ELBO, we propose to alternate between closed-form updates and stochastic gradient descent steps thanks to the results presented in the next section.

Analytical updates with respect to the model parameters $\boldsymbol{\tau}$, γ, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ The first-order conditions applied to the ELBO with respect to $\boldsymbol{\tau}$ and the model parameters give closed-form updates as stated in the following proposition.

Proposition 1. *Let $\mathscr{L}(R(\cdot);\Theta)$ denote the ELBO described in Equation* (12)*. The first-order conditions with respect to τ, γ, $(\mu_q)_q$ and $(\sigma_q)_q$ give the following updates:*

$$
\tau_{iq} = \frac{\gamma_q e^{-\mathrm{KL}_{iq}}}{\sum_{r=1}^{Q} \gamma_r e^{-\mathrm{KL}_{ir}}}, \qquad (13)
$$

$$
\gamma_q = \frac{1}{N} \sum_{i=1}^{N} \tau_{iq}, \qquad (14)
$$

$$
\mu_q = \left(\sum_{i=1}^{N} \tau_{iq} \right)^{-1} \sum_{i=1}^{N} \tau_{iq} \mu_\phi(\mathbf{A})_i, \qquad (15)
$$

$$
\sigma_q^2 = \left((Q-1) \sum_{i=1}^{N} \tau_{iq} \right)^{-1} \sum_{i=1}^{N} \tau_{iq} \left((Q-1)\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2 \right). \qquad (16)
$$

One way to interpret the update of $\boldsymbol{\tau}$ is to note that the optimal probability of node cluster membership with respect to cluster q decreases exponentially fast with the Kullback-Leibler divergence between the variational distribution of Z_i and the distribution of the

representation of cluster q. The update of γ corresponds to the approximated posterior expectation of the cluster proportions in the network. On the one hand, the optimal μ_q is given by the posterior means of $(Z_i)_i$ weighted by each node contribution to the corresponding cluster. On the other hand, the optimal variance σ_q^2 is given by the sum of two terms. The first one corresponds to the weighted mean of the posterior variances of the nodes. The second one corresponds to the weighted mean of the squared Euclidean distances between the node posterior means and μ_q. In other words, the variances incorporate both the uncertainty about the posterior variance, illustrated by the $\sigma_\phi^2(A)_i$ terms, as well as the uncertainty regarding the μ_ϕ, corresponding to the $\|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2$ terms.

Stochastic gradient descent One of the core difficulties in this model is the estimation of the parameters $\mathbf{\Pi}$ and the variational parameters ϕ due to the intractable term $\mathbb{E}_R\left[\log p(A \mid \mathbf{Z}, \mathbf{\Pi})\right]$, and in particular $\mathbb{E}_R\left[\log(\sum_{q,r} \eta_{i,q}\eta_{j,r}\pi_{qr})\right]$. To overcome this issue, we rely on a stochastic gradient descent algorithm using the reparametrisation trick (Kingma and Welling, 2014; Rezende et al., 2014) enabling easy computations of the gradient estimates with low variances.

4 Evaluation on synthetic datasets

In real-life datasets, quantifying the relevance of a network representation as well as node partitions is a challenging task since no partition of the node exists. Therefore, to assess Deep-LPBM ability to cluster the data, it is necessary to compare its clustering results with a ground truth on synthetic data. First, we present the network structures used in this section to evaluate our methodology. Second, we illustrate the information provided by Deep-LPBM results on the challenging disassortative structures. Third, a comparison of Deep-LPBM representational capacity with the variational graph auto-encoder (VGAE, Kipf and Welling, 2016) and the deep latent position model (Liang et al., 2022:Deep-LPM,) on the three proposed connectivity structures is exposed. To end this section, we give a quantitative assessment of the clustering results on the community, the hub and the disassortative structures and compare our results with Deep-LPM clustering and the K-means algorithm applied on VGAE embeddings.

4.1 Presentation of network structures and Deep-LPBM results

To assess Deep-LPBM capacity to represent different network topologies, we evaluate our methodology on three different structures, all composed of 100 nodes and 5 clusters. The **community structure**, where nodes in the same cluster have a high probability of connection set to 0.5, while nodes in different clusters have a low probability of connection set to 0.01. The **disassortative structure** where nodes in different cluster have a high probability of connection set to 0.5, while nodes in the same cluster have a low probability of connection set to 0.01. Finally, we also use a **hub structure** which is a combination of both with one cluster following the disassortative pattern and the four others being communities.

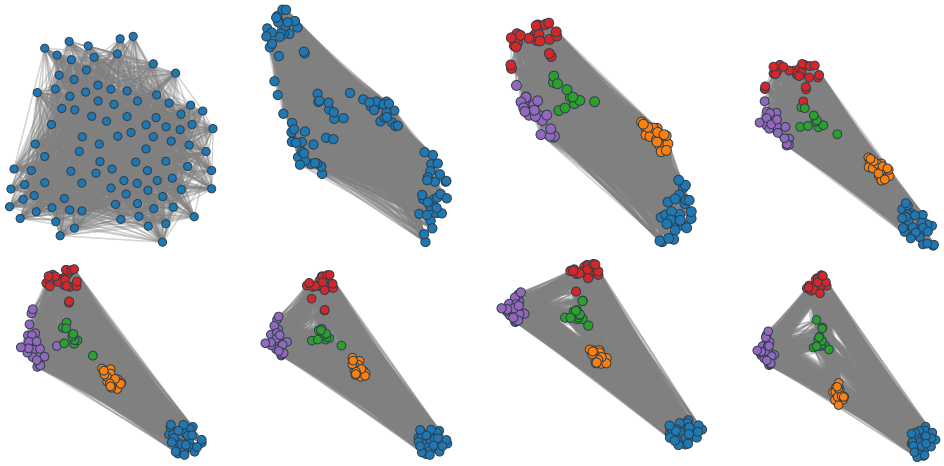

Figure 1: Evolution of the node embeddings during the estimation of Deep-LPBM on a disassortative network. The networks at the top (at the bottom respectively) correspond, from left to right, to the embeddings at the start of the GCN initialisation, the end of the GCN initialisation, iteration 100 and iteration 200 of Deep-LPBM (iteration 500, 1000, 1500, 2000 respectively). The embeddings were projected in $\mathbb{R}^2$ using the t-sne algorithm.

4.1.1 Learning node representations

These two sections highlight the flexibility of Deep-LPBM by analysing both the block modelling estimates as well as the network representation. We start with the latter, with the evolution of the representation during the optimisation presented in Figure 1.

The results provided in this section are obtained by fitting Deep-LPBM on a disassortative network and projecting the estimated embeddings in $\mathbb{R}^2$ with a t-sne algorithm (Van der Maaten and Hinton, 2008). First, we observe an efficient separation of the clusters which cannot be obtained with a similarity-based decoder since the probability of connection would increase with the correlation of the node embeddings. Therefore, the latent space would not be able to show any structure, as depicted in Figure 3. On the contrary, the model we propose is capable of imposing a structure on the variational distribution such as to obtain a node latent space matching with the connectivity patterns of the network.

4.1.2 Block modelling information

Although positional models offer a visualisation of the entire network, meta-representation of a network, such as Figure 2 can only be obtained by a block modelling strategy. The connectivity structure of the graph, captured by the matrix $\mathbf{\Pi}$, is displayed in Figure 2 as well as the associated meta-graph. A meta-graph is a network composed of nodes representing the estimated clusters with a size proportional to the estimated cluster proportions γ. The edge widths of the meta-graph are proportional to $\mathbf{\Pi}$. In large networks, this type of representation presents the advantage of being easily interpretable as well as focusing on the generative understanding of the network. Here, it is clear that the clusters are highly connected one to another but nodes from the same cluster are poorly connected together.

	1	2	3	4	5
1	0.01	0.5	0.51	0.52	0.49
2	0.5	0.0	0.48	0.54	0.52
3	0.51	0.48	0.0	0.49	0.52
4	0.52	0.54	0.49	0.0	0.5
5	0.49	0.52	0.52	0.5	0.02

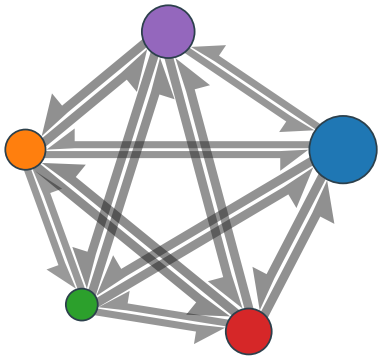

Figure 2: Meta-network based on Deep-LPBM results with an underlying disassortative structure. On the left-hand side, we provide the estimation of the connection probability matrix $\mathbf{\Pi}$. On the right-hand side, the meta-network is composed of nodes representing the clusters, their size is proportional to the corresponding estimated cluster proportion γ and the edge widths are proportional to $\mathbf{\Pi}$. A threshold has been set such that edges corresponding to a probability of connection lesser or equal to 0.02 are not displayed.

4.2 Representational power on three network structures

In Figure 3, both VGAE and Deep-LPBM efficiently render a latent space capturing the community structure as well as the hub structure. Eventually, the disassortative structure is more difficult to represent. Let us recall that the VGAE decoder models the probability of connection between two nodes with a sigmoid function applied to the cosine similarity between their respective latent positions. Therefore, it necessarily fails to capture the disassortative structure of the network. Hence, two nodes in the same cluster, that have a low probability of connection cannot be represented similarly, and thus cannot be close in the latent space. Conversely, Deep-LPBM is able to translate the connectivity pattern into the position of the nodes, such that nodes of the same cluster, poorly connected together, are close in the latent space.

4.3 Clustering evaluation on synthetic data

In this section, we aim to assess the clustering performance of our methodology. We compare it with Deep-LPM that relies on node embedding similarity as a decoder as well as VGAE used to estimate the node embeddings followed by K-means algorithm fitted on these embeddings. The benchmark is performed on networks with 100 nodes and $Q = 5$ clusters. The results are displayed in Table 1. First, we note the efficiency of Deep-LPBM on communities and hubs, reaching an ARI of 1 in both cases. It does not degrade the good performance of its competitors on these architectures, with ARIs of 1 and 0.97 for the VGAE and K-means, and an ARI of 1 on both structures for Deep-LPM. However, these competitors are not able to represent any connectivity structure in the disassortative case. In particular, as shown in Figure 3 for the VGAE, they cannot find relevant node clusters in this setting and end up with an ARI of 0.02 and 0. On the contrary, our methodology reaches an ARI of 0.8, much

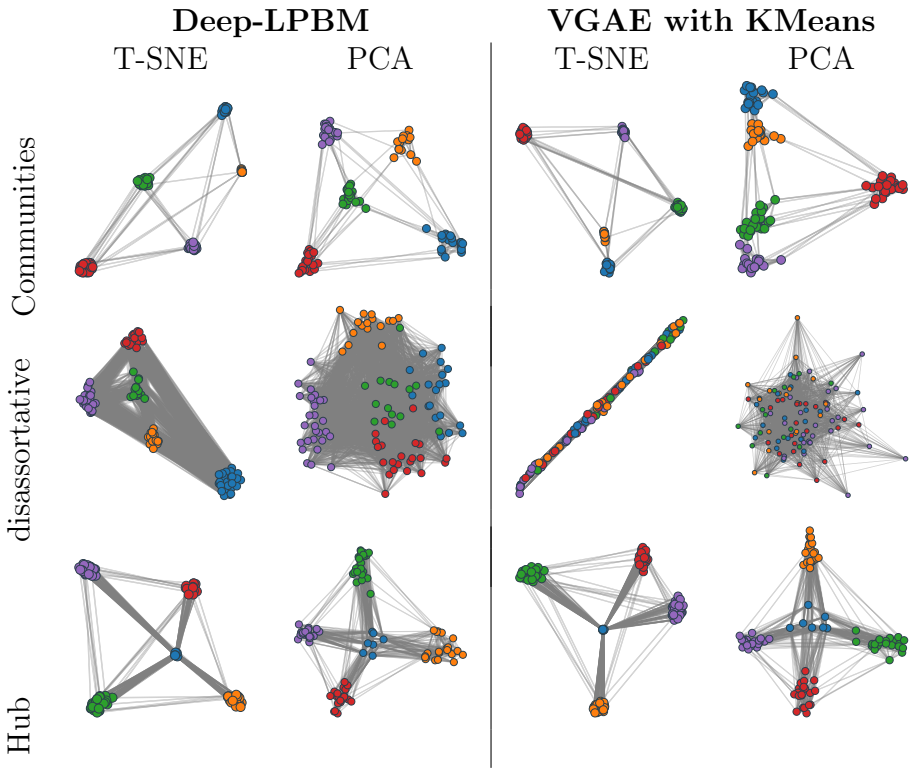

Figure 3: Example of three networks with a latent structure composed of (top to bottom) five communities, a disassortative network with five clusters and four communities with a hub. On the left-hand side (right-hand side respectively), the networks represent the results obtained fitting Deep-LPBM (VGAE) and using a t-sne projection (left) as well as a PCA (right). Each node colour corresponds to its true cluster membership.

higher than its competitors.

5 Conclusions

This paper introduced a new methodology combining a block model with a deep latent position model. By modifying the edge distribution and marginalising over a bijective transformation of the node latent representations, we managed to use the node embeddings as cluster probability memberships. We obtained richer results, providing a high-level meta-network, as well as a full network representation, to incorporate details at the node-level. Deep-LPBM is based on the encoder of a graph variational autoencoder combined with a novel block model decoder. Experiments showed that on communities, hubs and disassortative networks, our methodology rightfully translated the network salient information into the latent space. In addition, the clustering results are competitive with the state-of-the-art Deep-LPM. This is an on going work and a more extensive benchmark will be provided if the paper were to be accepted.

	Communities	disassortative	Hub
VGAE + Kmeans	1.00 ± 0.01	0.02 ± 0.02	0.97 ± 0.06
Deep-LPM	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
Deep LPBM	1.00 ± 0.00	0.80 ± 0.08	1.00 ± 0.00

Table 1: ARI obtained by a K-means algorithm applied on VGAE node embeddings, Deep-LPM and Deep-LPBM partitions. We keep the best initialisation ELBO wise over 10 initialisations and repeat it over 10 network to obtain the means and standard deviations.

References

Daudin, J-J, Franck Picard, and Stéphane Robin (2008). *A mixture model for random graphs*. In: Statistics and computing, Vol. 18, No. 2, pp. 173–183.

Daudin, Jean-Jacques, Laurent Pierre, and Corinne Vacher (2010). *Model for heterogeneous random networks using continuous latent variables and an application to a tree–fungus network*. In: Biometrics, Vol. 66, No. 4, pp. 1043–1051.

Handcock, Mark S, Adrian E Raftery, and Jeremy M Tantrum (2007). *Model-based clustering for social networks*. In: Journal of the Royal Statistical Society: Series A (Statistics in Society), Vol. 170, No. 2, pp. 301–354.

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

Hoff, Peter D, Adrian E Raftery, and Mark S Handcock (2002). *Latent space approaches to social network analysis*. In: Journal of the american Statistical association, Vol. 97, No. 460, pp. 1090–1098.

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

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

Liang, Dingge et al. (2022). *Deep latent position model for node clustering in graphs*. In: The 30th European Symposium on Artificial Neural Networks (ESANN 2022).

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan 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.

Snijders, Tom AB and Krzysztof Nowicki (1997). *Estimation and prediction for stochastic blockmodels for graphs with latent block structure*. In: Journal of classification, Vol. 14, No. 1, pp. 75–100.

Van der Maaten, Laurens and Geoffrey Hinton (2008). *Visualizing data using t-SNE*. In: Journal of machine learning research, Vol. 9, No. 86, pp. 2579–2605.

Xu, Zhiqiang, Yiping Ke, and Yi Wang (2014). *A Fast Inference Algorithm for Stochastic Blockmodel*. In: 2014 IEEE International Conference on Data Mining, pp. 620–629.