doi: 10.1093/biomet/asab058

Latent space models for multiplex networks with shared structure

BY P. W. MACDONALD, E. LEVINA AND J. ZHU

Department of Statistics, University of Michigan,
1085 South University, Ann Arbor, Michigan 48109, U.S.A.

pwmacdon@umich.edu elevina@umich.edu jizhu@umich.edu

SUMMARY

Latent space models are frequently used for modelling single-layer networks and include many popular special cases, such as the stochastic block model and the random dot product graph. However, they are not well developed for more complex network structures, which are becoming increasingly common in practice. In this article we propose a new latent space model for multiplex networks, i.e., multiple heterogeneous networks observed on a shared node set. Multiplex networks can represent a network sample with shared node labels, a network evolving over time, or a network with multiple types of edges. The key feature of the proposed model is that it learns from data how much of the network structure is shared between layers and pools information across layers as appropriate. We establish identifiability, develop a fitting procedure using convex optimization in combination with a nuclear-norm penalty, and prove a guarantee of recovery for the latent positions provided there is sufficient separation between the shared and the individual latent subspaces. We compare the model with competing methods in the literature on simulated networks and on a multiplex network describing the worldwide trade of agricultural products.

Some key words: Latent space model; Multilayer nework; Multiplex network.

1. INTRODUCTION

Network data have become commonplace in many statistical applications, such as in neuroscience, the social sciences and computational biology. In the vast majority of cases, these network data are represented as graphs. At a minimum, a graph $G = (V, E)$ has a node set V and an edge set E, with each edge connecting a pair of nodes, but frequently additional information is available, such as node attributes, edge weights and multiple types of edges. Although a lot of work has been done on a single network with binary edges, as the complexity of the network data structure increases, the availability of statistical methods and models dwindles rapidly. There is a strong need for rigorous statistical analysis to keep up with the rapidly increasing complexity of real datasets.

One such complex network data structure is the multilayer graph (Kivelä et al., 2014), a highly general mathematical object that can describe multiple graphs, dynamic graphs, hypergraphs, and vertex-coloured or edge-coloured graphs. In addition to a node set and an edge set, a multilayer graph includes a layer set. A node may appear on any or all layers, and each edge connects two vertices, including the possibility of connection in the same layer, called an intra-layer edge; across layers, called an inter-layer edge; and between the same node in different layers. For example, a general multilayer network could be used to represent a multimodal urban transportation network of bus, train, bicycle and other connections, where each layer corresponds to a different mode of transportation and edges define connections between stations.

The focus of this paper is on multiplex graphs, a type of multilayer graph in which a common set of n nodes appears on every layer and no inter-layer edges are allowed. For example, the brain connectivity networks of a sample of people or a multi-commodity international trade network could be represented as a multiplex network where each layer corresponds to a subject or a commodity, respectively.

For a single undirected graph G with $|V(G)| = n$, a common modelling approach is to assume that there are n latent variables $\{X_i\}_{i=1}^n \subseteq \mathcal{X}$, one for each node. Typically, one further assumes that for each node pair $i \leqslant j$, X_i and X_j fully parameterize the distribution of the edge variable $E_{ij} = \mathbb{1}\{(i,j) \in E(G)\}$, and all the edge variables are mutually independent (Matias & Robin, 2014). The latent positions themselves are sometimes treated as fixed and sometimes treated as independent random variables; in the latter case, the above assumptions are conditional on $\{X_i\}_{i=1}^n$. These models are called latent space models, and intuitively the latent variable X_i represents the behaviour of node i through its position in the latent space $\mathcal{X}$. Matias & Robin (2014) distinguish between two cases: a discrete latent space $\mathcal{X} = \{1, \ldots, K\}$, so that each node is in one of K latent classes; and $\mathcal{X} = \mathbb{R}^d$, so that each node is represented by its coordinates in Euclidean space. The first case includes the ubiquitous stochastic block model (Holland et al., 1983), and a well-studied example of the second case is the random dot product graph (Young & Scheinerman, 2007; Athreya et al., 2017). Models in the seminal papers of Hoff et al. (2002) and Handcock et al. (2007) correspond to the second case as well. Throughout this paper we focus on the second, continuous case and, following the convention in the literature, latent space will be understood to refer to Euclidean space.

Some of the frequentist approaches to latent space models treat the latent variables as random, and focus on estimation of and inference for the parameters governing their distribution(s), such as Bickel et al. (2013) in the stochastic block model setting. Many other works perform inference conditional on the latent variables and estimate them, especially when the goal is community detection, for example Lei & Rinaldo (2015) in the stochastic block model setting, Athreya et al. (2017) in the random dot product graph setting, and Ma et al. (2020) in a latent space model with edge covariates.

Some extensions of latent space models to multilayer networks have been proposed in the literature. These can be divided into two categories: general multiple networks, for instance from repeated measurements or multiple subjects; and dynamic or time-varying networks, for which the layers have a natural ordering. A review paper by Kim et al. (2018) details recent developments in the dynamic setting.

In the multiple-networks setting, latent space models with a Bayesian approach to estimation have been proposed by Gollini & Murphy (2016), Salter-Townshend & McCormick (2017), D'angelo et al. (2019) and Sosa & Betancourt (2021), among others. While hierarchical Bayesian approaches allow these models to adaptively share information or model dependence across layers, they tend to be computationally expensive for large networks.

For the larger networks we aim to work with, we focus on three recent frequentist approaches to latent space and low-rank modelling for multiple networks, using them as baselines with which to compare our proposed method. Arroyo et al. (2021) consider a collection of independent random dot product graphs with a common invariant subspace; that is, the expected adjacency matrices for each layer are assumed to share a common low-dimensional column space. This is similar to approaches taken by Levin et al. (2017), Nielsen & Witten (2018), Jones & Rubin-Delanchy (2021) and Wang et al. (2021). However, they do not consider the case where each layer also contains meaningful individual signals in addition to shared structure.

Zhang et al. (2020) consider a model in which expected adjacency matrices, after a logistic transformation, share a common low-rank structure. This framework allows for layer-specific parameters controlling degree heterogeneity, but no other individual structure.

Wang et al. (2019) aim to decompose each expected adjacency matrix into a common part and an individual part after applying a logistic transformation. They assume that the individual part is of low rank, but make no such assumptions on the common part. Thus, this method loses the interpretability afforded by the latent space approach, and has high variability unless there is a large number of layers.

Finally, our model bears a resemblance to other recent work that aims to summarize multiple matrix-valued observations outside of the networks setting. For example, Lock et al. (2020), in the setting of multiview data, proposed a joint and individual approach to matrix factorization; and De Vito et al. (2019) proposed a model for multi-study factor analysis that estimates both common and individual factors.

In extending latent space models to the multiple-networks setting, we seek a modelling approach that can leverage shared structure to improve estimation accuracy, but in an adaptive way, learning from the data how much the layers have in common instead of assuming that the entire latent representation is shared across all layers. We also allow for nontrivial individual structure in order to robustly estimate truly common structure.

As a motivating example, which will be analysed in § 6, we consider a multiplex network of international trade, where nodes correspond to countries, layers correspond to different commodities, and each weighted intra-layer edge is the total trade of a given agricultural commodity between two nations. We would expect the structure in this network to be governed by node attributes corresponding to, for instance, geographical region, language or climate. Some of these attributes would be expected to affect all commodities similarly; for example, geographical proximity should encourage trade of any commodity. On the other hand, some attributes may differ across layers; for example, climate may encourage production and hence trade of some commodities, but not others, depending on which crops are easiest to grow in a given country's climate. In a setting like this, if latent space models were fitted to each layer individually, then (i) a fitting procedure cannot leverage the shared structure across layers, and (ii) the latent representation of the common structure will not be automatically aligned across layers. On the other hand, if a single latent space model were fitted to all network layers jointly, or to some aggregated version, then (i) an influential individual latent dimension, or one that is shared by some, but not all layers, may be erroneously identified as a common effect; or (ii) the influence of a common latent dimension may be overstated if it is not orthogonal to the individual latent dimensions. The model we propose in the next section aims to address these shortcomings.

2. A new model for multiplex networks

2.1. *Multiplex networks with shared structure*

We propose a new model for multiplex networks with shared structure, henceforth referred to as MultiNeSS, with the goal of ultimately learning the amount of shared structure from the data. We start by defining some notation. Suppose that we observe m undirected networks, weighted or unweighted, on a common set of n nodes with no self-loops. The networks are represented by their $n \times n$ adjacency matrices $\{A_k\}_{k=1}^m$. Each node i is associated with a fixed latent position describing its function in layer k, denoted by $x_{k,i} \in \mathbb{R}^{d_k}$. The edges are assumed to be independent conditional on these latent positions:

$$A_{k,ij} \overset{\text{ind}}{\sim} Q\{\cdot\,; \kappa(x_{k,i}, x_{k,j}), \phi\} \quad (i = 1, \dots, n;\ j = 1, \dots, n;\ i < j;\ k = 1, \dots, m),$$

where $Q(\cdot\,;\theta,\phi)$ is some edge entry distribution with a scalar parameter θ and possible nuisance parameters ϕ, and $\kappa(\cdot\,,\cdot)$ is a symmetric similarity function, implying that the parameter θ captures the effect of the latent similarity of nodes i and j in layer k on the corresponding edge. We denote the latent positions for layer k by X_k, where the ith row of the $n \times d_k$ matrix X_k corresponds to the latent position of node i in layer k. In general, the position and its dimension may depend on the layer k.

The choice of the similarity function κ may affect the identifiability of each X_k. For instance, if $\kappa(x,y) = \psi(x^{\mathrm{T}}y)$ is an invertible scalar function ψ applied to the Euclidean inner product, each X_k is identifiable only up to a common orthogonal rotation of the rows. If $\kappa(x,y) = \psi(\|x-y\|_2)$ is similarly defined as an invertible function of the Euclidean distance rather than the Euclidean inner product, then X_k is identifiable only up to a common orthogonal rotation and/or reflection of the rows, as well as a common shift of each row by a vector in $\mathbb{R}^{d_k}$.

The key assumption of the MultiNeSS model is that some, but not all, structure is shared across network layers. We suppose that the matrix X_k can be written as

$$X_k = \begin{pmatrix} V & U_k \end{pmatrix} \quad (k = 1, \ldots, m), \tag{1}$$

where $V \in \mathbb{R}^{n\times d_1}$ is a matrix of common latent position coordinates, and the $U_k \in \mathbb{R}^{n\times d_{2,k}}$ are individual latent position coordinates for layer k. Writing X_k in this way further complicates identifiability. The model will certainly be identifiable only up to some invariant transformation of the rows of each U_k and of V, but we would still want V to be identifiable in such a way that it is aligned across all the layers. Intuitively, for this to hold we need the common dimension d_1 to be maximal and unique, in the sense that any transformation which aligns the first d_1 coordinates must partition X_k into V and U_k as written above. We will formalize this intuition in § 2.2. First, we present some concrete examples of latent space models that fit the general MultiNeSS model framework.

Example 1 (Low rank, Gaussian errors). As a simple example, let the similarity function for each layer be the generalized inner product as described in Rubin-Delanchy et al. (2020). For vectors x and y in $\mathbb{R}^{p+q}$,

$$\kappa_{p,q}(x,y) = x_1y_1 + \cdots + x_py_p - x_{p+1}y_{p+1} - \cdots - x_{p+q}y_{p+q} = x^{\mathrm{T}}I_{p,q}y,$$

where $I_{p,q}$ is the block-diagonal matrix

$$I_{p,q} = \begin{pmatrix} I_p & 0 \\ 0 & -I_q \end{pmatrix},$$

with I_r for a positive integer r denoting the $r \times r$ identity matrix. Under the generalized inner product, the first p latent dimensions are referred to as assortative, while the remaining q are disassortative (Rubin-Delanchy et al., 2020).

Assume that $Q(\cdot\,;\theta,\sigma)$ is the Gaussian distribution $N(\theta,\sigma^2)$. Then each layer's adjacency matrix has expectation

$$E(A_k) = P_k = VI_{p_1,q_1}V^{\mathrm{T}} + U_kI_{p_{2,k},q_{2,k}}U_k^{\mathrm{T}} \quad (k = 1, \ldots, m),$$

where $p_1 + q_1 = d_1$, $p_{2,k} + q_{2,k} = d_{2,k}$ and each error matrix $E_k = A_k - E(A_k)$ is symmetric with independent and identically distributed zero-mean Gaussian entries.

If the setting does not allow self-loops, we can instead use

$$E(A_k) = P_k = VI_{p_1,q_1}V^{\mathrm{T}} + U_k I_{p_{2,k},q_{2,k}} U_k^{\mathrm{T}} \\ - \operatorname{diag}(VI_{p_1,q_1}V^{\mathrm{T}} + U_k I_{p_{2,k},q_{2,k}} U_k^{\mathrm{T}}) \quad (k = 1, \ldots, m)$$

to enforce zeros on the diagonal. The same can be done in any of the subsequent examples, if needed.

Example 2 (Low rank, exponential family errors). Let the similarity function κ be the generalized inner product again, and let $Q(\cdot\,;\theta)$ be a one-parameter exponential family distribution with natural parameter θ and log-partition function ν; that is,

$$Q(x;\theta) \propto \exp\{x\theta - \nu(\theta)\}.$$

For instance, Q may be a Bernoulli distribution, in which case $\nu(\theta) = \log\{1 + \exp(\theta)\}$. In the spirit of generalized linear models, we model the edges by applying the canonical link function $g = \nu'$ entrywise, so that adjacency matrices now satisfy

$$E(A_{k,ij}) = P_{k,ij} = g(v_i^{\mathrm{T}} I_{p_1,q_1} v_j + u_{k,i}^{\mathrm{T}} I_{p_{2,k},q_{2,k}} u_{k,j}) \\ (i = 1, \ldots, n;\ j = 1, \ldots, n;\ i \leqslant j;\ k = 1, \ldots, m).$$

In the Bernoulli example, the canonical link function is the inverse logistic function

$$g(\theta) = \exp(\theta)/\{1 + \exp(\theta)\}. \tag{2}$$

2.2. *Identifiability*

We present a sufficient condition for identifiability in the case where κ is a scalar function of the generalized inner product. For more detailed discussion of the statistical implications of such transformations, see Rubin-Delanchy et al. (2020). For other choices of similarity function, conditions for identifiability will depend on the set of invariant transformations that it induces.

For the inner product model with one layer, it is natural to assume that the matrix of latent positions $X \in \mathbb{R}^{n\times d}$ is of full rank, i.e., it has linearly independent columns. We show that a stronger linear independence condition for all pairwise concatenations of the latent position matrices is sufficient for identifiability in the proposed MultiNeSS model. The proof is given in the Supplementary Material.

Proposition 1. *Suppose $\kappa_{p,q}(x, y) = \psi(x^{\mathrm{T}} I_{p,q} y)$ is an invertible scalar function of the generalized inner product, and that the model is parameterized by V and $\{U_k\}_{k=1}^m$ as in (1).*

Define an undirected graph $\mathcal{G}_I$ on the network layers, with vertex set $\{1, \ldots, m\}$ and edges

$$k \sim l \iff \begin{pmatrix} V & U_k & U_l \end{pmatrix} \text{ are linearly independent.} \tag{3}$$

If $\mathcal{G}_I$ is connected, then the model is identifiable up to indefinite orthogonal transformation. That is, if the probability distributions induced by two different parameterizations $(V, U_1, \ldots, U_m)$ and $(V', U_1', \ldots, U_m')$ coincide, then

$$V = V'W_0, \quad U_1 = U_1'W_1, \quad \ldots, \quad U_m = U_m'W_m$$

for some indefinite orthogonal transformations $\{W_k\}_{k=0}^m$.

To simplify the condition in Proposition 1, consider the special case in which $\mathcal{G}_I$ is the complete graph, equivalent to assuming that for all $1 \leqslant k_1 < k_2 \leqslant m$, the $n \times (d_1 + d_{2,k_1} + d_{2,k_2})$ matrix $\begin{pmatrix} V & U_{k_1} & U_{k_2} \end{pmatrix}$ has linearly independent columns. In the special case where $q_1 = q_{2,1} = \cdots = q_{2,m} = 0$, the similarity function for the latent vectors is the standard Euclidean inner product, and Proposition 1 holds with identifiability up to orthogonal rotation.

If we assume that the fully concatenated $n \times (d_1 + \sum_k d_{2,k})$ matrix $\begin{pmatrix} V & U_1 & \cdots & U_m \end{pmatrix}$ has linearly independent columns, as in De Vito et al. (2019) for a similar factor analysis model, then once again $\mathcal{G}_I$ will be the complete graph and Proposition 1 will hold. Proposition 1 does not require orthogonality of the columns of V and $\{U_k\}_{k=1}^m$, although clearly it will hold if the columns are all mutually orthogonal.

As visual intuition, consider a simple case with $n = 10$, $d_1 = d_2 = 1$ and $m = 2$. Standard results for the random dot product graph (Athreya et al., 2017) suggest that the two-dimensional latent positions for each layer are identifiable only up to orthogonal rotations, which differ across layers. The recovered latent positions for the two layers may have different rotations, and so would not share a common column according to the MultiNeSS model (1). Proposition 1 states that pairwise linear independence is sufficient to uniquely align the rotations, and identify the common and individual latent positions up to sign.

In Fig. 1 panels (a) and (b) we plot latent positions $\{(v_i, u_{k,i})\}_{i=1}^n$ in $\mathbb{R}^2$ for $k = 1$ and 2, respectively. Each point is labelled with its index, 1–10, for ease of matching across the panels. The common dimension is on the x-axis and the individual dimension on the y-axis; thus the x-coordinates are the same in panels (a) and (b). In Fig. 1 panels (c) and (d) we apply an orthogonal transformation to each of the latent positions, which is equivalent to applying an unknown two-dimensional orthogonal rotation. After rotation, the points in panel (c) do not match the points in panel (d) in either their x- or their y-coordinates. The dashed lines denote the original x-axis in the two rotated spaces; observe that the coordinates of projection onto these directions are constant in panels (a) and (b). After rotation we identify two directions, indicated by dashed lines in panels (c) and (d), with the property that for all points the coordinates of projection onto these directions are the same, up to a sign flip, in panels (c) and (d). By Proposition 1, as long as (3) holds, the original x-axis is the unique direction with this property, and the coordinates of projection uniquely identify the entries of v, up to a sign flip.

3. Fitting the MultiNeSS model

3.1. *Convex objective function*

A natural approach to latent space estimation is likelihood maximization. A convex relaxation of the likelihood can be maximized by introducing a nuclear-norm penalty and optimizing over the entries of the low-rank matrices $F = VI_{p_1,q_1}V^{\mathrm{T}}$, and $G_k = U_k I_{p_{2,k},q_{2,k}} U_k^{\mathrm{T}}$ for each $k = 1, \ldots m$ rather than the latent position matrices themselves.

With the notation defined in § 2, suppose κ is a generalized inner product on $\mathbb{R}^d$. In terms of the latent position parameters $(V, \{U_k\}_{k=1}^m)$, the negative loglikelihood, after dropping constants, takes the form

$$\ell(V, \{U_k\}_{k=1}^m \mid \{A_k\}_{k=1}^m) \propto -\sum_{k=1}^m \sum_{i \leqslant j} \log Q(A_{k,ij}; v_i^{\mathrm{T}} I_{p_1,q_1} v_j + u_{k,i}^{\mathrm{T}} I_{p_{2,k},q_{2,k}} u_{k,j}, \phi), \quad (4)$$

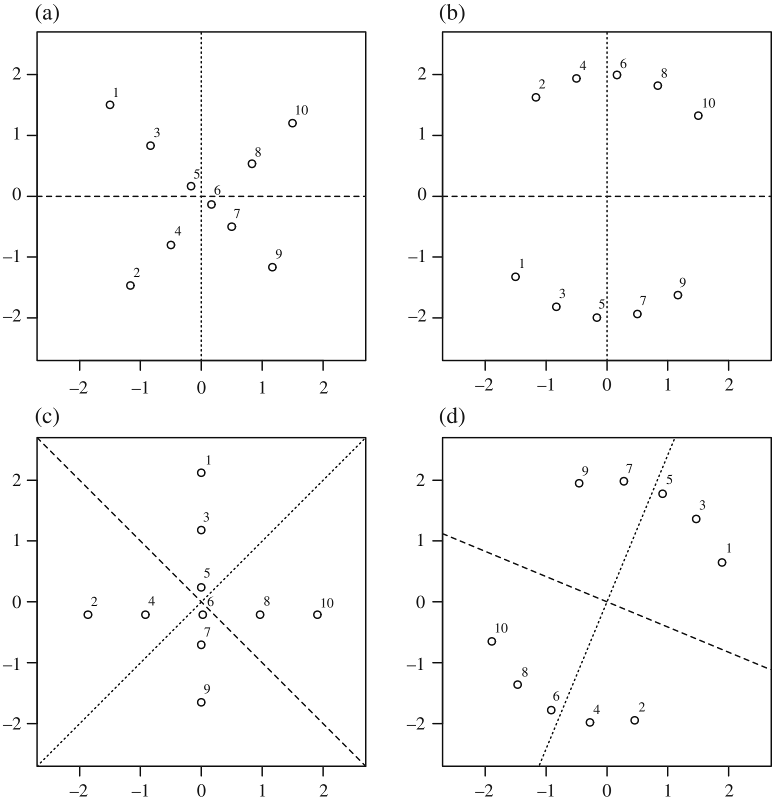

Fig. 1. Latent positions (a) in layer 1 before rotation, (b) in layer 2 before rotation, (c) in layer 1 after rotation, and (d) in layer 2 after rotation. Points are labelled according to their index from 1 to 10.

where Q is the density of the edge weight distribution. Up to a rotation, we can rewrite this likelihood in terms of symmetric $n \times n$ matrices $F = VI_{p_1,q_1}V^{\mathrm{T}}$ and $G_k = U_k I_{p_{2,k},q_{2,k}} U_k^{\mathrm{T}}$ $(k = 1,\ldots m)$ by constraining the number of positive and negative eigenvalues of each matrix. For a symmetric matrix M, let $\mathrm{r}^+(M)$ and $\mathrm{r}^-(M)$ denote, respectively, the numbers of strictly positive and strictly negative eigenvalues of M. Then we equivalently minimize

$$\ell(F, \{G_k\}_{k=1}^m \mid \{A_k\}_{k=1}^m) = -\sum_{k=1}^{m} \sum_{i \leqslant j} \log Q(A_{k,ij}; F_{ij} + G_{k,ij}, \phi)$$

subject to the constraints

$$\begin{aligned} &\mathrm{r}^+(F) \leqslant p_1, \quad \mathrm{r}^+(G_k) \leqslant p_{2,k} \quad (k = 1,\ldots m), \\ &\mathrm{r}^-(F) \leqslant q_1, \quad \mathrm{r}^-(G_k) \leqslant q_{2,k} \quad (k = 1,\ldots m). \end{aligned}$$

When κ is the Euclidean inner product, $q_1 = q_{2,1} = \cdots = q_{2,m} = 0$, and the constraints are equivalent to requiring each matrix to be of low rank and positive semidefinite.

To make this problem tractable, we ignore the constraint on the eigenvalue signs and perform a further convex relaxation of the resulting rank constraint, leading to the unconstrained convex optimization problem

$$\min_{F,\,G_k} \left\{ \ell(F, \{G_k\}_{k=1}^m \mid \{A_k\}_{k=1}^m) + \lambda \|F\|_* + \sum_{k=1}^m \lambda\alpha_k \|G_k\|_* \right\}, \tag{5}$$

where $\lambda \geqslant 0$ and $\alpha_k \geqslant 0$ $(k = 1, \ldots, m)$ are tuning parameters, and $\|\cdot\|_*$ denotes the nuclear norm of a matrix, i.e., the sum of the singular values. The parameter λ appears in both terms as an overall scaling that depends on n and the total entrywise variance across all the layers, while each α_k controls the individual penalties, depending on the entrywise variance of each layer. Since the nuclear norm is convex, it is easy to see that (5) defines a convex optimization problem as long as the edge distribution Q is log-concave in θ.

3.2. *Proximal gradient descent algorithm*

The optimization problem can be solved by applying proximal gradient descent blockwise to each of the matrix arguments. In particular, we split the optimization variables into $m + 1$ blocks of n^2 variables: one block containing the entries of F, and m blocks, one for the entries of each G_k.

Then for each block the negative loglikelihood is convex and differentiable, and the nuclear-norm penalty term is convex and, though nondifferentiable, has a well-defined proximal mapping for step size $\eta > 0$ (Fithian & Mazumder, 2018). In particular, the nuclear norm scaled by $\lambda \geqslant 0$ has the proximal mapping

$$\arg\min_{M'} \frac{1}{2\eta} \|M - M'\|_{\mathrm{F}}^2 + \lambda \|M'\|_* = S_{\eta\lambda}(M),$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the matrix Frobenius norm, i.e., the Euclidean norm of the vectorized entries, and $S_T(\cdot)$ is the soft singular-value thresholding operator with threshold $T \geqslant 0$; that is, for a diagonal matrix $M \in \mathbb{R}^{q\times q}$,

$$S_T(M) = \mathrm{diag}\{(M_{11} - T)_+, \ldots, (M_{qq} - T)_+\},$$

and otherwise $S_T(M) = U S_T(D) V^{\mathrm{T}}$, where $M = UDV^{\mathrm{T}}$ is the singular-value decomposition of M (Fithian & Mazumder, 2018).

Thus, we derive the following proximal gradient descent steps with step size η/m for updates of F and step size η for updates of each G_k: at iteration step $t \geqslant 1$,

$$\hat{F}^{(t)} = S_{\eta\lambda/m} \left\{ \hat{F}^{(t-1)} + \frac{\eta}{m} \frac{\partial}{\partial F} \ell(\hat{F}^{(t-1)}, \{\hat{G}_{k'}^{(t-1)}\}_{k'=1}^m) \right\},$$

$$\hat{G}_k^{(t)} = S_{\eta\lambda\alpha_k} \left\{ \hat{G}_k^{(t-1)} + \eta \frac{\partial}{\partial G_k} \ell(\hat{F}^{(t)}, \{\hat{G}_{k'}^{(t-1)}\}_{k'=1}^m) \right\} \quad (k = 1, \ldots, m).$$

This particular choice of relative step sizes is discussed in the Supplementary Material.

When Q is a one-parameter exponential family and the edge distribution is modelled through the canonical link function as in Example 2, the gradients take a particularly nice form. In particular, up to an additive constant, for each fixed node pair (i,j),

$$(\log Q)'(A_{k,ij}; F_{ij} + G_{k,ij}) = A_{k,ij} - E(A_{k,ij}; F_{ij} + G_{k,ij}) \quad (k = 1, \ldots, m),$$

where the equality follows from the choice of link function $g = \nu'$. Hence, the gradients with respect to each G_k can be interpreted as the residual from estimating the adjacency matrix by its expectation, given the current low-rank parameters. The gradient with respect to F is the sum of the residuals over all the layers.

If $Q(\cdot\,;\theta,\phi)$ is the Gaussian distribution as in Example 1, the appropriate link function is the identity link. Then with step size $1/m$ for updates of F and step size 1 for updates of each G_k, proximal gradient descent recovers a natural alternating soft-thresholding algorithm: at iteration step $t \geqslant 1$,

$$\hat{F}^{(t)} = S_{\lambda/m}\left\{\frac{1}{m}\sum_{k=1}^{m}(A_k - \hat{G}_k^{(t-1)})\right\}, \quad \hat{G}_k^{(t)} = S_{\lambda\alpha_k}\big(A_k - \hat{F}^{(t)}\big) \quad (k = 1, \ldots, m). \tag{6}$$

In § 4 we will provide theoretical guarantees on estimators found with this special case of proximal gradient descent. When the observed networks have no self-loops, we perform proximal gradient steps that ignore the diagonal entries, which should yield better empirical results in this case.

Although we have presented the algorithm for the generalized inner product similarity, it can easily be adapted to the usual inner product similarity by enforcing a positive-semidefinite constraint on F and on each G_k in (5). The constraint can be enforced by adding a positive-semidefinite projection step at each iteration of the proximal gradient descent algorithm, equivalent to shrinking the negative eigenvalues of each iterate to zero.

The most computationally expensive part of each update is the singular-value decomposition needed for soft singular-value thresholding. If the full singular-value decomposition is calculated, each iteration step has a computational complexity of $O(mn^3)$. In practice, we use a truncated singular-value decomposition that only finds the first $s \ll n$ singular vectors and values, as in Wu et al. (2017), reducing the complexity to $O(mn^2 s)$. For synthetic multiplex networks generated according to the models from Examples 1 and 2 with $n = 400$ and $m = 8$, as in § 5, our `R` (R Development Core Team, 2022) implementation of proximal gradient descent is able to perform approximately one iteration per second. When the signals are sufficiently strong, the algorithm typically converges in fewer than 10 steps.

3.3. *Refitting step*

As we will show in Theorem 1, recovery of the correct rank requires a tuning parameter of order $\lambda \sim \surd n$, and thus the effect of the soft-thresholding step on the estimated eigenvalues will not disappear as $n \to \infty$.

As in Mazumder et al. (2010), we propose a refitting step after solving the convex problem, where we fix the ranks and eigenvectors of the estimated $\hat{F}$ and $\hat{G}_k$, and refit their eigenvalues to maximize the original nonconvex likelihood.

Based on the output from the first step, we write the eigendecompositions

$$\hat{F} = \tilde{V}\hat{\Gamma}_F\tilde{V}^{\mathrm{T}}, \quad \hat{G}_k = \tilde{U}_k\hat{\Gamma}_k\tilde{U}_k^{\mathrm{T}} \quad (k = 1, \ldots m). \tag{7}$$

Elementwise, we have

$$\hat{F}_{ij} = \sum_{\ell=1}^{\hat{d}_1}\gamma_\ell(\hat{F})\tilde{V}_{i\ell}\tilde{V}_{j\ell} \quad (i = 1, \ldots, n;\, j = 1, \ldots n),$$

where $\hat{d}_1$ is the rank of $\hat{F}$ and $\gamma_\ell(\hat{F})$ denotes the ℓth eigenvalue of $\hat{F}$, ordered by magnitude. The elements of each $\hat{G}_k$ can be expressed similarly, with $\hat{d}_{2,k}$ denoting the rank of $\hat{G}_k$. Then, fixing the estimated eigenvectors, the refitting step solves the convex problem

$$\min_{\hat{\Gamma}_F,\hat{\Gamma}_k}\left[-\sum_{k=1}^{m}\sum_{i\leqslant j}\log Q\left\{A_{k,ij};\sum_{\ell=1}^{\hat{d}_1}\gamma_\ell(\hat{F})\tilde{V}_{i\ell}\tilde{V}_{j\ell}+\sum_{\ell=1}^{\hat{d}_{2,k}}\gamma_\ell(\hat{G}_k)\tilde{U}_{k,i\ell}\tilde{U}_{k,j\ell},\phi\right\}\right]. \qquad (8)$$

When Q is a one-parameter exponential family and the edge distribution is modelled through the corresponding canonical link function as in Example 2, solving (8) is exactly equivalent to fitting a generalized linear model with $n(n+1)m/2$ responses and $\hat{d}_1+\sum_{k=1}^{m}\hat{d}_{2,k}$ predictors.

With the solution to the refitting step problem (8), we can construct the final estimates for the low-rank matrices based on these refitted eigenvalue estimates, along with the original estimated eigenvectors defined in (7).

3.4. *Choosing tuning parameters*

A standard method for choosing tuning parameters is cross-validation, which requires some care when performed on networks. We take an approach motivated by the edge cross-validation method for networks proposed by Li et al. (2020), where a random subsample of node pairs is repeatedly removed, a low-rank matrix completion method is applied to the adjacency matrix to impute the missing pairs, and the original method is refitted on the completed matrix. Tuning parameters are then selected to minimize a loss function evaluated on the held-out node pairs.

Whereas the general edge cross-validation procedure of Li et al. (2020) contains an imputation step followed by a fitting step, the MultiNeSS fitting approach can be applied directly to adjacency matrices with missing entries. Suppose we subsampled matrices $\{A_k\}_{k=1}^m$ by removing the values for a random sample of indices (i,j,k), accounting for symmetry. Denote the set of remaining indices by Ω. The new loglikelihood will resemble (4), but with the summation restricted to the triples in Ω, and the same proximal gradient descent algorithm can be applied.

Similar to the approach taken by Lock et al. (2020) for low-rank multiview data matrices, the tuning parameters can also be chosen adaptively using random matrix theory. In particular, in Example 1 with known σ constant across all layers, bounds on the singular values of $\sum_{k=1}^{m}E_k$ would suggest setting $\lambda=(2+\delta)\sigma(nm)^{1/2}$ for a constant δ. Gavish & Donoho (2014) introduced an estimator $\hat{\sigma}_{\mathrm{MAD}}$ for σ based on the median singular value and suggested setting $\delta=0.309$, which is optimal for hard singular-value thresholding. However, δ could also be selected using edge cross-validation. Then, a constant σ across layers suggests the choice $\alpha_k=m^{-1/2}$ $(k=1,\ldots,m)$. This adaptive tuning scheme is used for the evaluation on synthetic networks in § 5. While this approach is designed with Example 1 in mind, it gives sensible results in Example 2 with Bernoulli edges as well when the networks are sufficiently dense. For sparse networks with Bernoulli edges, we recommend setting $\lambda=C(nm)^{1/2}$ and $\alpha_k=m^{-1/2}$ $(k=1,\ldots,m)$, where C is a constant selected using edge cross-validation.

This adaptive tuning approach can also be used to account for layer-specific variances. Suppose $\hat{\sigma}^2_{\mathrm{MAD}}(A_k)$ estimates the entrywise variance for layer k. Then, rather than taking α_k to be the same for all layers, we set it based on the relative variance estimates for the different layers:

$$\alpha_k=\left\{\frac{\hat{\sigma}_{\mathrm{MAD}}(A_k)^2}{\sum_{k'=1}^{m}\hat{\sigma}_{\mathrm{MAD}}(A_{k'})^2}\right\}^{1/2}\qquad(k=1,\ldots,m).$$

As above, λ is selected based on the singular values of $\sum_{k=1}^{m} E_k$,

$$\lambda = (2+\delta)\sqrt{n}\left\{\sum_{k=1}^{m} \hat{\sigma}_{\mathrm{MAD}}(A_k)^2\right\}^{1/2}, \tag{9}$$

where again δ is a constant that is either chosen a priori or selected using edge cross-validation. This layer-specific adaptive tuning is used for the real data analysis in § 6.

The estimation algorithms described in this section for the models in Examples 1 and 2, including options for refitting and parameter tuning, are implemented in an `R` package `multiness`, available at `github.com/peterwmacd/multiness`.

4. Theoretical guarantees

4.1. *Notation*

We denote the matrix ℓ_2 operator norm by $\|M\|_2$. Let

$$[M]_d = \operatorname*{arg\,min}_{M' :\, \mathrm{rank}(M') \leqslant d} \|M - M'\|_{\mathrm{F}},$$

which is well-defined, by the Eckart–Young theorem, as the truncation of the singular-value decomposition of M to the largest d singular values. For $d, p, q \geqslant 0$, let $\mathcal{O}_d$ denote the set of $d \times d$ rotation (orthonormal) matrices, and let $\mathcal{O}_{p,q}$ denote the set of $(p+q) \times (p+q)$ indefinite orthogonal matrices. Let $\mathrm{col}(M)$ and $\mathrm{row}(M)$ denote the column and row spaces of a matrix M, respectively. For a symmetric matrix M, let $\gamma_i(M)$ denote the ith eigenvalue of M, with eigenvalues ordered from largest to smallest in absolute value. Any reference to the leading or first eigenvalue of a symmetric matrix refers to the largest eigenvalue in absolute value.

4.2. *Main results*

Throughout this section we assume the model described in Example 1, where $Q(\cdot\,; \theta, \sigma)$ is the Gaussian distribution with known variance σ^2. To simplify notation, we assume that d_2 is constant in k, although the results generalize to the case where $d_{2,k}$ can depend on k, replacing d_2 in the assumptions by $\max_k d_{2,k}$. We allow the dimensions n, m, d_1 and d_2 to grow, subject to the following restrictions.

Assumption 1. We have that $d_2 2^{d_2} m^2 n^{1-2\tau} = o(1), d_1 m^{-1} = o(1), d_2 m^{-1} = o(1)$ for some constant $\tau \in (1/2, 1]$.

Assumption 1 places bounds on the total number of latent dimensions relative to the number of nodes n. We study the estimator of the MultiNeSS model, defined as the limit of the proximal gradient update steps (6), starting from some initial value $\hat{F}^{(0)}$. Let $\hat{F}$ and $\{\hat{G}_k\}_{k=1}^{m}$ denote the limits of this proximal gradient descent algorithm as $t \to \infty$.

As in (7), let $G_k = \bar{U}_k \Gamma_k \bar{U}_k$ $(k = 1, \ldots, m)$ denote the eigendecomposition of each G_k and $F = \bar{V} \Gamma_F \bar{V}^{\mathrm{T}}$ the eigendecomposition of F. Suppose that they satisfy the following assumptions.

Assumption 2. We have that

$$\begin{aligned} b_1 n^{\tau} &\leqslant |\gamma_{d_2}(G_k)| \leqslant |\gamma_1(G_k)| = \|G_k\|_2 \leqslant B_1 n^{\tau} \quad (k = 1, \ldots, m), \qquad (10)\\ b_1 n^{\tau} &\leqslant |\gamma_{d_1}(F)| \leqslant |\gamma_1(F)| = \|F\|_2 \leqslant B_1 n^{\tau} \end{aligned}$$

for uniform constants $0 < b_1 \leqslant B_1$. Further,

$$\|\bar{V}^{\mathrm{T}}\bar{U}_k\|_2 = o(d_1^{-1/2}m^{1/2}n^{1/2-\tau}) \quad (k = 1, \ldots, m), \tag{11}$$

$$\|\bar{U}_{\mathcal{A}}^{\mathrm{T}}\bar{U}_k\|_2 \leqslant B_2\sigma|\mathcal{A}|^{1/2}n^{1/2-\tau} \quad (k = 1, \ldots, m) \tag{12}$$

for some uniform constant $B_2 > 0$, where $\mathcal{A} \subseteq \{1, \ldots, m\} \setminus \{k\}$ and $\bar{U}_{\mathcal{A}}$ is an orthonormal basis for the subspace sum $\sum_{j\in\mathcal{A}} \mathrm{col}(G_j)$. In particular,

$$\|\bar{U}_{k_1}^{\mathrm{T}}\bar{U}_{k_2}\|_2 \leqslant B_2\sigma n^{1/2-\tau} \quad (k_1 = 1, \ldots, m;\ k_1 = 1, \ldots, m;\ k_1 \neq k_2). \tag{13}$$

Although stated with fixed orthonormal bases, (11)–(13) are basis-free and can be written in terms of the maximal cosine similarity between elements of the two column spaces. That is, if S_1 and S_2 are two subspaces of $\mathbb{R}^n$, then for any of their respective orthonormal bases U_{S_1} and U_{S_2},

$$\|U_{S_1}^{\mathrm{T}}U_{S_2}\|_2 = \sup_{x\in S_1, y\in S_2} \frac{|x^{\mathrm{T}}y|}{\|x\|_2\|y\|_2}.$$

Comparing (11) and (13), we observe that these conditions allow for slightly more similarity between the column spaces of F and any one G_k than between the column spaces of G_k and G_j for $k \neq j$.

Assumption 2 controls the signal strength through the eigenvalues of F and each G_k, and controls the separation between the common and individual latent dimensions through bounds on the inner products of eigenvectors of F and each G_k. As our framework treats the latent positions as deterministic, we make assumptions directly about these eigendecompositions rather than about the generative distribution of the latent positions.

Under these assumptions we have the following consistency result. The proof is given in the Supplementary Material.

Theorem 1. *Suppose $Q(\cdot\,;\theta,\sigma) = N(\theta,\sigma^2)$ and that Assumptions 1 and 2 hold. Let $\lambda = 3c_\lambda\sigma(nm)^{1/2}$ and $\alpha_k = (c_\lambda\sqrt{m})^{-1}$ $(k = 1, \ldots, m)$, where c_λ is a universal constant. Then with probability greater than $1 - (m+1)n\exp(-C_0 n)$ for some universal constant $C_0 > 0$, the initializer*

$$\hat{F}^{(0)} = \left[\frac{1}{m}\sum_{k=1}^{m} A_k\right]_{d_1}$$

satisfies

$$\|\hat{F}^{(0)} - F\|_{\mathrm{F}} = o(n^{1/2}), \tag{14}$$

and for n sufficiently large and for all $k \in \{1, \ldots, m\}$, we have

$$n^{-1}\|\hat{F} - F\|_{\mathrm{F}} \leqslant C_1\sigma d_1^{1/2}(nm)^{-1/2}, \quad n^{-1}\|\hat{G}_k - G_k\|_{\mathrm{F}} \leqslant C_2\sigma d_2^{1/2}n^{-1/2} \quad (k = 1, \ldots, m) \tag{15}$$

for positive constants C_1 and C_2 that do not depend on n, m, d_1, d_2 and σ. Moreover, if all the eigenvalues of F and each G_k are nonnegative, then $\hat{F}$ and each $\hat{G}_k$ are positive semidefinite.

Remark 1. The initializer $\hat{F}^{(0)}$ uses the true value of d_1, which is generally unknown in practice. However, since the objective is convex, the estimators should not be sensitive to the initial value.

Remark 2. The conditions of Theorem 1 provide a regime under which our convex approach achieves the same rate as an oracle hard-thresholding approach. In particular, if we estimated each G_k with full knowledge of F and estimated F with full knowledge of each G_k by

$$\hat{F}^{(\text{oracle})} = \left[\frac{1}{m}\sum_{k=1}^{m}(A_k - G_k)\right]_{d_1}, \quad \hat{G}_k^{(\text{oracle})} = [A_k - F]_{d_2} \quad (k = 1, \ldots, m),$$

they would have the same Frobenius-norm error rates as the estimators in Theorem 1.

Remark 3. In the proof of Theorem 1, we bound the operator norm of each error matrix E_k using a concentration inequality for Gaussian random matrices (Bandeira & Van Handel, 2016). With a different operator-norm concentration inequality (Chatterjee, 2015), we can show that a similar result holds if the entries of E_k are uniformly bounded instead of Gaussian. For instance, this would yield consistency for a random dot product graph-like binary edge model with $F + G_k \in [0, 1]^{n\times n}$ for $k = 1, \ldots, m$ and

$$A_{k,ij} \sim \text{Ber}(F_{ij} + G_{k,ij}) \quad (i = 1, \ldots, n;\ j = 1, \ldots, n;\ i < j;\ k = 1, \ldots, m).$$

Although Assumption 1 allows us to match the oracle error rate, it also places a strong requirement on the latent dimensions, especially the individual latent dimension d_2. Theorem 2 gives an alternative result under a weaker assumption on d_2, when it is allowed to grow polynomially in n. The proof is given in the Supplementary Material.

Assumption 3. We have that $m^2 n^{1-2\tau} = o(1)$, $d_2 d_1 m^{-1} = o(1)$ for some constant $\tau \in (1/2, 1]$.

THEOREM 2. *Suppose* $Q(\cdot\,; \theta, \sigma) = N(\theta, \sigma^2)$ *and that Assumptions* 2 *and* 3 *hold. Let* $\lambda = 3c_\lambda \sigma (d_2 nm)^{1/2}$ *and* $\alpha_k = \{c_\lambda (d_2 m)^{1/}\}^{-1}$ $(k = 1, \ldots, m)$, *where* c_λ *is a universal constant. Then with probability greater than* $1 - (m + 1)n \exp(-C_0 n)$ *for some constant* $C_0 > 0$, *the initializer*

$$\hat{F}^{(0)} = \left[\frac{1}{m}\sum_{k=1}^{m} A_k\right]_{d_1}$$

satisfies

$$\|\hat{F}^{(0)} - F\|_{\text{F}} = o(n^{1/2}),$$

and for n sufficiently large, we have

$$n^{-1}\|\hat{F} - F\|_{\text{F}} \leqslant C_3 \sigma (d_1 d_2)^{1/2} (nm)^{-1/2},$$
$$n^{-1}\|\hat{G}_k - G_k\|_{\text{F}} \leqslant C_4 \sigma d_2^{1/2} n^{-1/2} \quad (k = 1, \ldots, m)$$

for positive constants C_3 *and* C_4 *that do not depend on n, m,* d_1, d_2 *and* σ. *Moreover, if all the eigenvalues of* F *and each* G_k *are nonnegative, then* $\hat{F}$ *and each* $\hat{G}_k$ *are positive semidefinite.*

Theorems 1 and 2 provide bounds on the recovery of the $n \times n$ matrix-valued parameters F and G_k; however, in practice we are often interested in the latent position matrices V and U_k as well. With an additional assumption on the eigenvalue gaps of F and each G_k, the following proposition establishes overall consistency for an adjacency spectral embedding-based estimate of the latent positions, after a suitable linear transformation.

Since in general $\hat{F}$ and each $\hat{G}_k$ may have negative eigenvalues, we define the adjacency spectral embedding based on the absolute values of the eigenvalues as in Rubin-Delanchy et al. (2020). For instance, denoting by $\hat{F} = \tilde{V}\hat{\Gamma}_F\tilde{V}^{\mathrm{T}}$ the truncated eigendecomposition up to rank d_1 of $\hat{F}$, we define the d_1-dimensional adjacency spectral embedding of $\hat{F}$ by $\hat{V} = \tilde{V}|\hat{\Gamma}_F|^{1/2}$.

Assumption 4. We have that $\min_{j\in\{2,\ldots,d_1\}} \left\{|\gamma_j(F)| - |\gamma_{j-1}(F)|\right\} \geqslant b_3 n^{\xi}$ for some $\xi \in (1/2, \tau]$ and positive constant b_3, and an analogous condition holds for the eigenvalues of each G_k matrix with the same constant ξ.

This assumption on the eigenvalue gaps ensures that the ordering of latent dimensions is preserved in the estimates of F and of each G_k. We have the following consistency result for the latent matrices V and each U_k, up to rotation. The proof is given in the Supplementary Material.

Proposition 2. *Suppose Theorem 1 and Assumption 4 hold. Then with probability greater than* $1 - (m+1)n\exp(-C_0 n)$ *for some universal constant* $C_0 > 0$ *and for sufficiently large* n, $\hat{F}$ *and each* $\hat{G}_k$ *are low-rank matrices. Further, let* $\hat{V}$ *be the* $(n \times d_1)$*-dimensional adjacency spectral embedding of* $\hat{F}$*, and let* $\hat{U}_k$ *be the* $(n \times d_2)$*-dimensional adjacency spectral embedding of* $\hat{G}_k$ *for each* $k = 1, \ldots, m$*. Let* p_1 *and* q_1 *denote the numbers of assortative and disassortative common latent dimensions, respectively, so that* $F = VI_{p_1,q_1}V^{\mathrm{T}}$*. Define* p_2 *and* q_2 *similarly. Then*

$$(d_1 n)^{-1/2} \inf_{W\in\mathcal{O}_{p_1,q_1}} \|\hat{V} - VW\|_{\mathrm{F}} \leqslant C_5\sigma d_1^{1/2} m^{-1/2} n^{\tau/2-\xi}, \tag{16}$$

$$(d_2 n)^{-1/2} \inf_{W\in\mathcal{O}_{p_2,q_2}} \|\hat{U}_k - U_k W\|_{\mathrm{F}} \leqslant C_6\sigma d_2^{1/2} n^{\tau/2-\xi} \quad (k = 1, \ldots, m) \tag{17}$$

for some positive constants C_5 *and* C_6.

Remark 4. Since we assume $\xi > 1/2 \geqslant \tau/2$, Proposition 2 shows that under the asymptotic regime of Assumption 1, the average entrywise error of the latent position matrices, after suitable linear transformation, goes to zero. As in Theorem 1, the rate of convergence for the common structure exceeds that of the individual structure by a factor of $\surd m$.

5. Evaluation on synthetic networks

5.1. *Baseline methods*

Throughout this section we compare the estimator for the MultiNeSS model with baseline methods on two types of synthetic networks, those with weighted edges generated according to the Gaussian model in Example 1, and those with binary edges generated according to the logistic model in Example 2. We compare the proposed method with nonadaptive optimization approaches for the MultiNeSS model and with other methods for multiple networks (Wang et al., 2019; Arroyo et al., 2021) that can capture the common or individual low-rank structure.

We also include two nonconvex oracle approaches in our comparison. For the Gaussian model, we apply a nonconvex alternating rank truncation algorithm that assumes oracle knowledge of the true ranks d_1 and d_2. The alternating updates for $t \geqslant 1$ are given by

$$\hat{F}^{(t)} = \left[\frac{1}{m}\sum_{k=1}^{m}\left(A_k - \hat{G}_k^{(t-1)}\right)\right]_{d_1}, \quad \hat{G}_k^{(t)} = \left[A_k - \hat{F}^{(t)}\right]_{d_2} \quad (k = 1, \ldots, m)$$

and are initialized with $\hat{G}_k^{(0)} = 0$ $(k = 1, \ldots, m)$. These update steps are applied until convergence or until a prespecified maximum iteration number $t_{\max}$ is reached.

For the logistic model, we compare our convex approach with a nonconvex gradient descent algorithm, similar to that of Ma et al. (2020), which also assumes known d_1 and d_2. This procedure directly updates the entries of the latent position matrices V and each U_k by performing gradient descent on the negative loglikelihood function.

The COSIE method recently proposed by Arroyo et al. (2021) fits a low-rank model to multiple binary undirected networks on a common node set. This method provides estimates of the expected adjacency matrices for each layer, but does not decompose the estimates into common and individual parts, so we can only compare the accuracy of the overall expectation. Although COSIE is designed for the random dot product graph model, it can also be applied unchanged to the Gaussian model. We use an oracle version assuming knowledge of the true d_1 and d_2. For a fair comparison with our method, we first identify the $d_1 + d_2$ leading eigenvectors for each layer and then fit a common invariant subspace of dimension $d_1 + md_2$, the total number of latent dimensions in the MultiNeSS model.

The second baseline method under comparison is the M-GRAF algorithm proposed by Wang et al. (2019) for a similar logistic link model for multilayer networks with common and individual parts. The M-GRAF model does not assume any structure, low-rank or otherwise, for entries of the common matrix F and does not employ regularization; it is thus better suited to the regime with small n and large m. We apply an oracle version that assumes knowledge of the true individual rank d_2. Since M-GRAF does not assume a common low-rank structure, it does not need a value for d_1.

5.2. *Gaussian model results*

We consider instances of the Gaussian model with no self-loops and the usual inner product similarity. We set $d_1 = d_2 = 2$ and $\sigma = 1$, and we vary $n \in \{200, 300, 400, 500, 600\}$ with fixed $m = 8$ and vary $m \in \{4, 8, 12, 15, 20, 30\}$ with fixed $n = 400$. In each setting we generate 100 independent realizations from the model. The entries of the common and individual latent position matrices are generated as independent standard normals, so while they are not strictly orthogonal, their expected correlation is zero. Under the Gaussian model we compare four methods: the MultiNeSS estimator with and without the refitting step, denoted by MultiNeSS and MultiNeSS+, respectively; the alternating rank truncation approach, referred to as Nonconvex; and the COSIE method of Arroyo et al. (2021).

We evaluate the methods on how well they do in recovering the common structure, the individual structure and the overall expectation of the adjacency matrix, using relative Frobenius-norm errors with $\|\cdot\|_{\tilde{\mathrm{F}}}$ denoting the Frobenius norm that ignores diagonal entries:

$$\mathrm{Err}_F = \frac{\|\hat{F} - F\|_{\tilde{\mathrm{F}}}}{\|F\|_{\tilde{\mathrm{F}}}}, \quad \mathrm{Err}_G = \frac{1}{m}\sum_{k=1}^{m}\frac{\|\hat{G}_k - G_k\|_{\tilde{\mathrm{F}}}}{\|G_k\|_{\tilde{\mathrm{F}}}},$$
$$\mathrm{Err}_P = \frac{1}{m}\sum_{k=1}^{m}\frac{\|\hat{F} + \hat{G}_k - F - G_k\|_{\tilde{\mathrm{F}}}}{\|F + G_k\|_{\tilde{\mathrm{F}}}}. \tag{18}$$

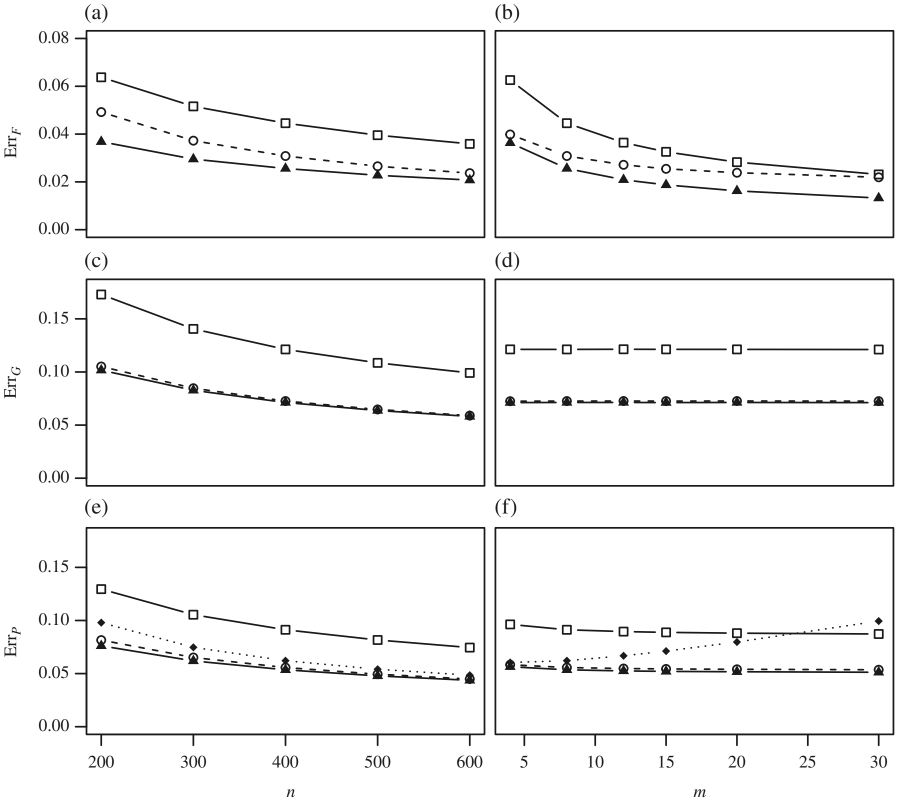

Fig. 2. Frobenius-norm errors for the common structure (top row), individual structure (middle row) and overall expected value (bottom row) under the Gaussian model. The left panels plot the errors against the number of nodes n, with a fixed number of layers $m = 8$. The right panels plot the errors against the number of layers m, with a fixed number of nodes $n = 400$. The methods under comparison are MultiNeSS (□, solid) , MultiNeSS+ (▲, solid), Nonconvex (∘, dashed) and COSIE (⧫, dotted).

The results are shown in Fig. 2. Several general conclusions can be drawn. MultiNeSS without the refitting step does not outperform the nonconvex oracle, but MultiNeSS+ is uniformly the best method in all cases, though the nonconvex oracle performs very similarly in estimating the individual layers G_k. One possible explanation for the improvement over the nonconvex oracle is that the convex optimization approach ignores the diagonal elements of the adjacency matrices, which do not reflect the true low-rank structure.

All the methods perform better as the number of nodes n grows, as we would expect. Increasing the number of layers m has no effect on errors in estimating the individual components for MultiNeSS, since each one is estimated separately, but it helps us estimate F better by pooling shared information across more layers and therefore also improves the overall estimation of P. The rate of decrease of the error in F seems to match well the rate of $m^{-1/2}$ predicted by the theory. COSIE, on the other hand, benefits from increasing n, but suffers when m grows, with the overall error in P going up with m. We conjecture that this happens because COSIE must first estimate a subspace of dimension $d_1 + md_2$, which leads to high variability as m increases.

Comparing panel (e) to panels (a) and (c), and comparing panel (f) to panels (b) and (d), we see that the estimation error for P_k is on average less than the estimation error for G_k, implying that the error in P_k does not decompose additively into the error in G_k and the error in F. Even when the expected correlation in the latent position matrices is zero, it is challenging to correctly distinguish common structure from individual structures.

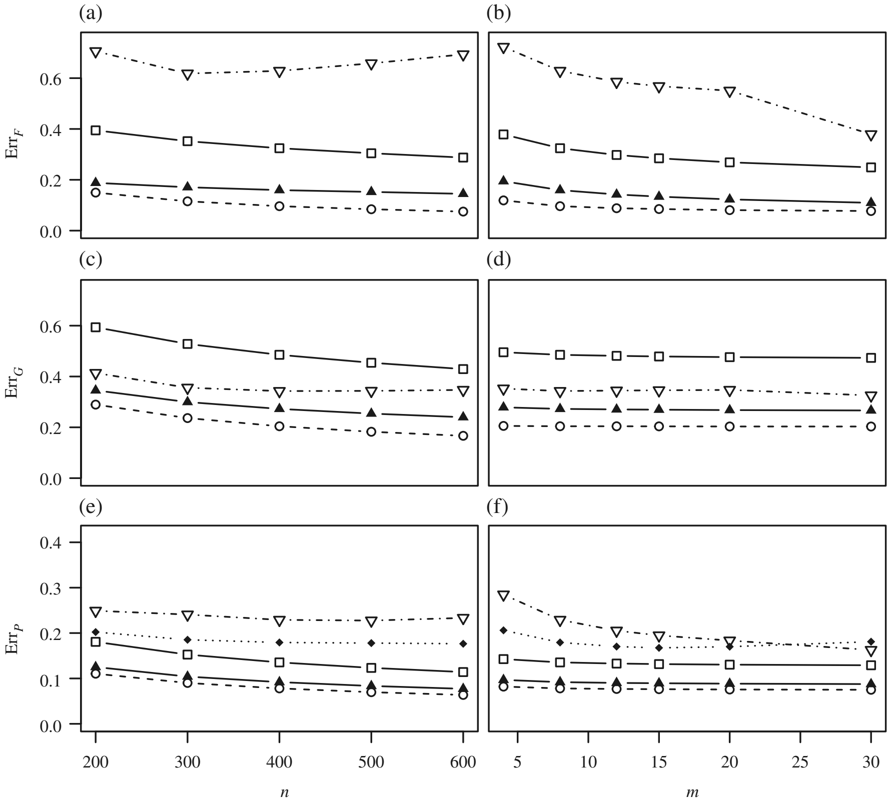

Fig. 3. Frobenius-norm errors for the common structure (top row), individual structure (middle row) and overall expected value after the logistic transformation (bottom row) under the logistic model. The left panels plot the errors against the number of nodes n, with a fixed number of layers $m = 8$. The right panels plot the errors against the number of layers m, with a fixed number of nodes $n = 400$. The methods under comparison are MultiNeSS (□, solid), MultiNeSS+ (▲, solid), Nonconvex (○, dashed), COSIE (♦, dotted) and M-GRAF (▽, dot-dash).

5.3. *Logistic model results*

We also consider instances of the logistic model with no self-loops, the same inner product similarity, and $d_1 = d_2 = 2$, where we vary $n \in \{200, 300, 400, 500, 600\}$ with fixed $m = 8$ and vary $m \in \{4, 8, 12, 15, 20, 30\}$ with fixed $n = 400$. In each setting we generate 100 independent realizations of the model. The entries of the common and individual latent position matrices are generated as independent standard normals. We compare five methods: the MultiNeSS estimator with and without the refitting step, again denoted by MultiNeSS and MultiNeSS+, the nonconvex approach, COSIE and M-GRAF. COSIE does not use the correct model for these data, since it assumes a random dot product graph model without a logistic link. We evaluate the recovery of the common and individual structures using the same relative Frobenius-norm errors given in (18). To evaluate the overall recovery of the expected value for each layer, we use the relative Frobenius-norm error after elementwise application of the inverse logistic link function; that is, we redefine

$$\mathrm{Err}_P = \frac{1}{m}\sum_{k=1}^{m} \frac{\|g(\hat{F} + \hat{G}_k) - g(F + G_k)\|_{\tilde{\mathrm{F}}}}{\|g(F + G_k)\|_{\tilde{\mathrm{F}}}},$$

where g is as defined in (2).

The results are shown in Fig. 3. Many of the general conclusions in this example are the same as for the Gaussian model. M-GRAF does not perform better as n increases, and performs much worse for small values of m, since it does not regularize the common matrix F in any way. In contrast to the Gaussian model, here the nonconvex approach slightly outperforms MultiNeSS+. The difference between the nonconvex and MultiNeSS+ errors is driven by large-magnitude entries in F and G_k, which have a substantial effect on the log-odds scale, but little effect on the expectation of the adjacency matrix. Hence, the difference between these two methods is attenuated in panels (e) and (f) after applying the inverse logistic link function.

For the binary networks generated from the logistic MultiNeSS model, we also compare performance over a range of network edge densities by subtracting a density-controlling parameter $\beta \geqslant 0$ from the log-odds of each edge. As above, we generate V and $\{U_k\}_{k=1}^m$ as $n \times 2$ matrices of independent standard normals, resulting in $P_k = g(VV^{\mathrm{T}} + U_k U_k^{\mathrm{T}} - \beta \mathbb{1}_n \mathbb{1}_n^{\mathrm{T}})$. This is equivalent to generating networks from a logistic MultiNeSS model with generalized inner product similarity, augmenting the common latent position matrix with an extra disassortative latent dimension with coordinates $\beta^{1/2}\mathbb{1}_n$.

We consider instances of this logistic MultiNeSS model with no self-loops, $n = 400$, $m = 8$ and $\beta \in \{0, 1, 2, 3, 4, 5, 6\}$. MultiNeSS networks generated with these choices of β have edge densities of approximately 0.5, 0.34, 0.21, 0.12, 0.06, 0.035 and 0.015, respectively. We compare the MultiNeSS estimator with and without the refitting step, the nonconvex approach, COSIE and M-GRAF. In order to easily implement the nonconvex oracle approach, we fit it with full knowledge of d_1, d_2 and β. MultiNeSS without refitting is tuned adaptively with a fixed constant, as in the previous dense network simulations. MultiNeSS+ is tuned with edge cross-validation. The error Err_F for the recovery of the common structure is normalized by $\|VV^{\mathrm{T}}\|_{\mathrm{F}}^2$, ignoring the effect of β on the common structure; Err_P is calculated as above, including β in the normalizer.

The results are shown in Fig. 4. For highly sparse networks with edge densities of approximately 3.5% and 1.5%, M-GRAF does not converge consistently, so its results in these settings are omitted.

For edge densities greater than approximately 5%, the relative performances of the methods are similar to those seen for dense networks. As the edge density decreases, the nonconvex oracle unsurprisingly performs much better than MultiNeSS in panel (a), as it does not have to fit the density-controlling parameter β. For highly sparse networks with edge densities under 5%, we see that MultiNeSS+ outperforms MultiNeSS without refitting in panels (a) and (b), but has slightly worse error in panel (c). While MultiNeSS+ better controls the ranks of F and G_k, and more accurately recovers the latent coordinates, MultiNeSS without refitting performs best with a much smaller choice of λ, and can more accurately recover the expected adjacency matrix despite greatly overestimating the number of latent dimensions. Finally, we see that in the sparsest regime in panel (b), MultiNeSS+ outperforms the nonconvex oracle. In this case, MultiNeSS+ is able to adaptively ignore some weak latent dimensions, while the oracle nonconvex approach is forced to fit two individual latent dimensions per layer, even when the signal is too weak for its coordinates to be reliably estimated.

6. An agricultural trade network analysis

To illustrate the insights that can be gained from fitting a MultiNeSS model, we analyse a dataset of food and agriculture trade relationships between countries, collected in 2010. Each node corresponds to a country and each layer to a different agricultural product. The undirected edges are weighted by the bilateral traded quantity of the commodity. This dataset was previously analysed by De Domenico et al. (2015), who looked at structural similarities between layers.

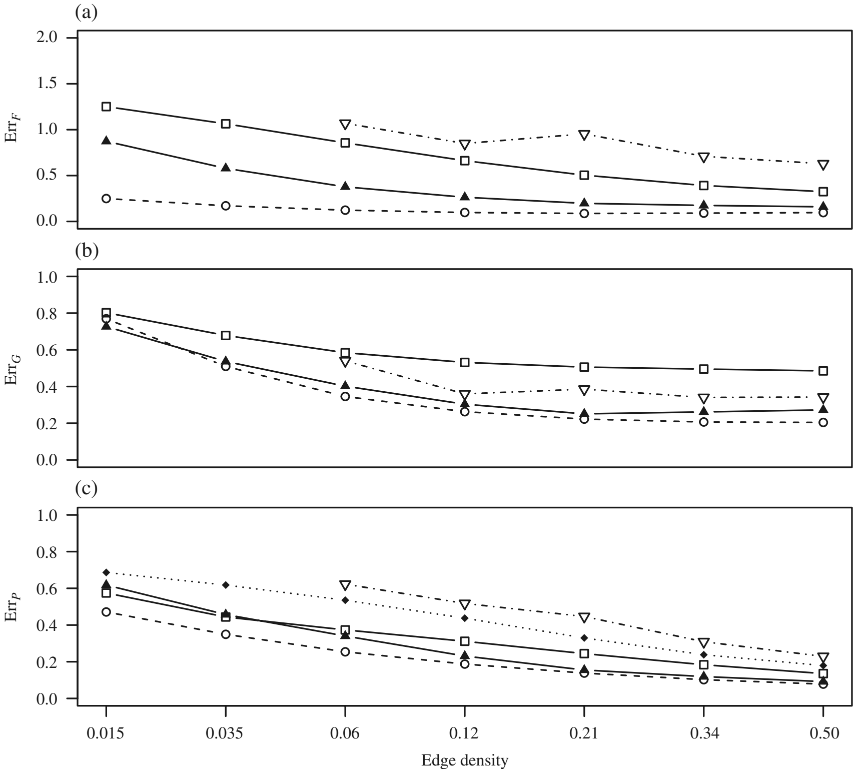

Fig. 4. Frobenius-norm errors for the common structure (top panel), individual structure (middle panel) and overall expected value (bottom panel) under the logistic model with varying edge density. The methods under comparison are MultiNeSS (□, solid), MultiNeSS+ (▲, solid), Nonconvex (○, dashed), COSIE (◆, dotted) and M-GRAF (▽, dot-dash).

As a pre-processing step, we remove low-density layers and nodes. The original dataset contains 214 countries and 364 products. We kept layers with at least 10% nonzero edges, and included nodes with a mean of at least five nonzero edges across these layers. The result is an undirected multiplex network containing no self-loops, with $n = 145$ nodes and $m = 13$ layers corresponding to agricultural products with high trade volume. Following common practice for this type of data, we use log trade volumes as edge weights, which also makes the assumption of Gaussian edge weights with constant variance within each layer more realistic.

We fit a Gaussian model using the MultiNeSS algorithm with refitting. The tuning parameters are selected using the layer-specific adaptive tuning approach described in § 3.4. The constant in (9) is set to $\delta = 1/2$ using edge cross-validation.

Figure 5 displays the results for the first four common latent dimensions, and Figs. 6 and 7 show the results for the first two individual dimensions for two example layers, wine and chocolate, respectively.

The estimated common matrix $\hat{F}$ has rank 39, with 25 assortative dimensions and 14 disassortative dimensions. Figure 5 shows scatter plots of the points projected onto the leading four latent dimensions, the first and second in panel (a), and the third and fourth in panel (b), which are all assortative. The first four singular values account for approximately 47% of the sum of the singular values of $\hat{F}$. The scatter plots suggest that the first latent dimension corresponds roughly to the total volume of trade and the subsequent dimensions correspond to regional trade relationships. In particular, the second dimension primarily separates Europe from Asia, the third separates the Americas from the rest of the world, and the fourth separates the Middle East and Africa from Asia and the Pacific.

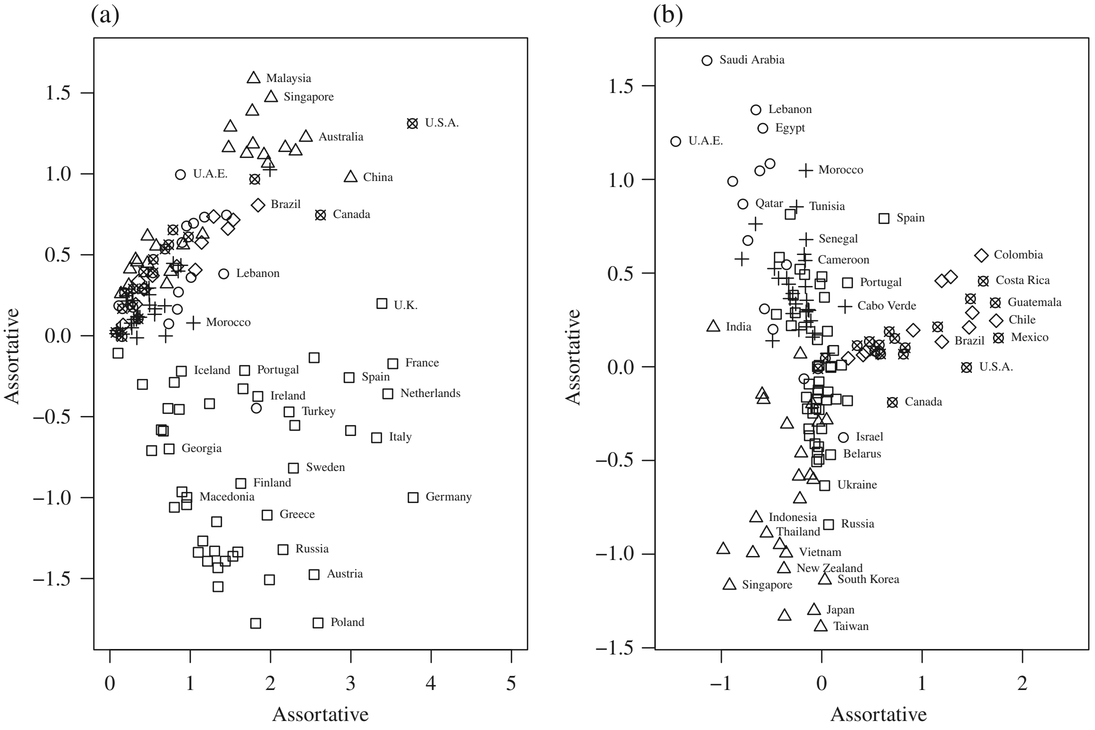

Fig. 5. Scatter plots of the first four MultiNeSS common latent dimensions for the food trade data: (a) dimensions 1 and 2; (b) dimensions 3 and 4. Dimensions are labelled assortative or disassortative. Points are drawn by geographical region: +, Africa; △, Asia-Pacific; □, Europe; ○, Middle East; ⊗, North America; ×, South America.

For the individual component of the wine trade layer, shown in Fig. 6, we estimate $\text{rank}(\hat{G}_{\text{wine}}) = 18$, with nine assortative and nine disassortative latent dimensions. We plot the coordinates of the first two latent dimensions, which account for about 37% of the sum of the singular values of $\hat{G}_{\text{wine}}$. The second latent dimension corresponds roughly to the total volume of wine production after correcting for the common structure, with countries such as France, Spain, Chile and New Zealand having very high scores, and majority-Muslim nations such as Saudi Arabia and Indonesia having very low scores. The first latent dimension is disassortative and assigns large positive coordinates to the major wine exporters who do not trade wine among themselves, separating them from major wine importers such as China.

For comparison, we also plot the countries projected onto the first two latent dimensions constructed by adjacency spectral embedding applied to just the wine layer of the trade network. We swap the order of the adjacency spectral embedding dimensions to ease visual comparison with the MultiNeSS embedding. While the adjacency spectral embedding dimensions have similar interpretations to those of MultiNeSS and lead to the same general conclusions about high-volume wine producers, the interpretation of the lower-left part of the scatterplot is much clearer in the MultiNeSS individual embedding.

For the chocolate trading network, we estimate the individual component rank as $\text{rank}(\hat{G}_{\text{chocolate}}) = 7$, with five assortative and two disassortative latent dimensions. Projections onto the first two individual latent dimensions, which account for about 48% of the sum of the singular values of $\hat{G}_{\text{chocolate}}$, are shown in Fig. 7. Overall, the pattern is similar to that in Fig. 6(a), with the two axes swapped. The first latent dimension gives chocolate-producing nations such as Switzerland and Belgium the highest scores, and countries such as Vietnam, which has a very low per-capita chocolate consumption, low scores. The second latent dimension is disassortative

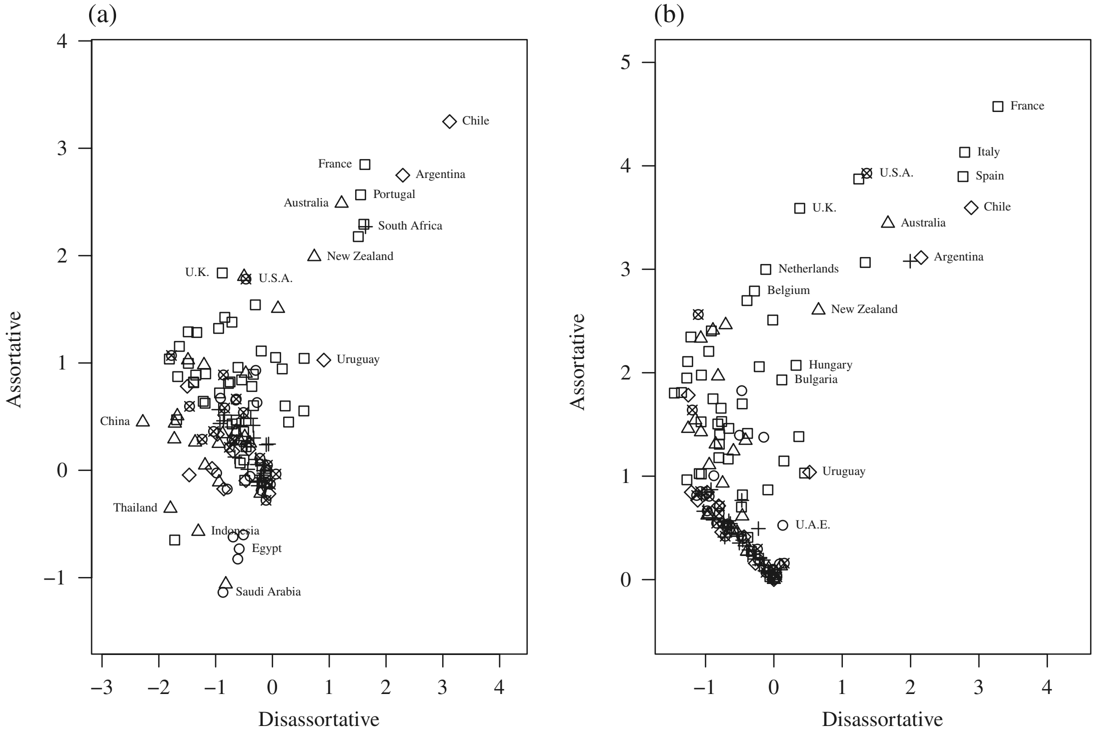

Fig. 6. Scatter plots of the first two individual latent dimensions for the wine layer: (a) dimensions 1 and 2 by MultiNeSS; (b) dimensions 1 and 2 by adjacency spectral embedding. Dimensions are labelled assortative or disassortative. Points are drawn by geographical region: +, Africa; △, Asia-Pacific; □, Europe; ○, Middle East; ⊗, North America; ×, South America.

and gives large positive coordinates to major chocolate exporters that do not trade chocolate with each other. Egypt and the U.A.E. also have outlying coordinates, as according to these data they primarily trade chocolate with other Middle Eastern nations rather than importing directly from Europe. The adjacency spectral embedding in Fig. 7(b) looks very similar to Fig. 5(a). Since this embedding does not account for the common structure, it primarily captures patterns common to all products, rather than the chocolate-specific patterns revealed by the MultiNeSS embedding.

7. Discussion

There are several directions in which we plan to take this work forward. Firstly, although in § 4 we establish consistency only for a Gaussian edge weight model, we expect that this can be extended to other well-behaved edge weight distributions. The models we have developed so far allow for only two kinds of latent dimensions: those which are individual to one layer, and those which are common to all layers. Extension to more structured models, where latent dimensions can be shared by some, but not all layers, would open up a larger range of applications. For example, we could impose a group structure on the layers, allowing for group effects and enabling an analogue of analysis-of-variance on networks. An example application in which this would be useful is neuroimaging, where brain connectivity networks of a treatment group and a control group of patients could be analysed jointly and the treatment effect estimated more accurately. These groups could also be learned from data, in a natural extension of this set-up to clustering.

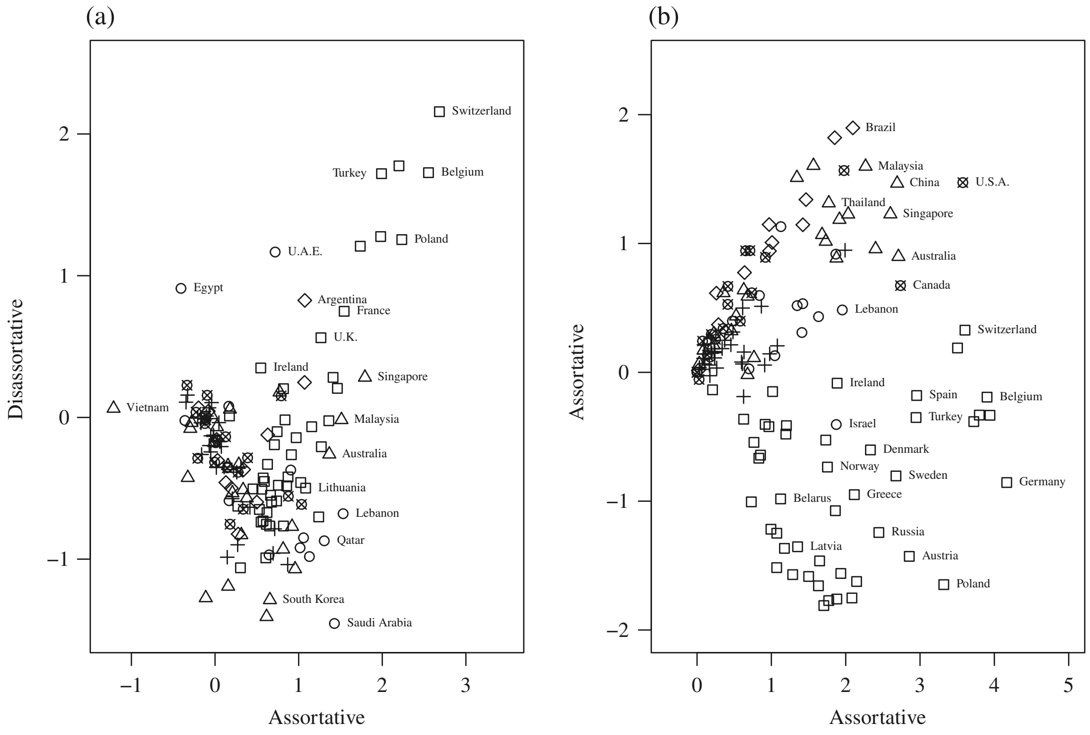

Fig. 7. Scatter plots of the first two individual latent dimensions for the chocolate layer: (a) dimensions 1 and 2 by MultiNeSS; (b) dimensions 1 and 2 by adjacency spectral embedding. Dimensions are labelled assortative or disassortative. Points are drawn by geographical region: +, Africa; △, Asia-Pacific; □, Europe; ○, Middle East; ⊗, North America; ×, South America.

Another possible extension is to dynamic networks, where each layer represents a network snapshot at a discrete time-point. In this setting, the ordering of the layers matters. Latent dimensions could be modelled as constant over time or constant over a contiguous time window, with obvious applications to prediction and changepoint analysis. Finally, a highly interpretable latent structure could be obtained by imposing a tree structure on the latent dimensions, with shared latent dimensions between nodes determined by their last common ancestor on the tree.

While this work focuses on undirected networks, we also recognize the importance of extending the model to directed networks. In this case each node would have both incoming and outgoing coordinates for each latent dimension. For instance, we could model the common structure as $V_{\text{out}}V_{\text{in}}^{\text{T}}$ for $n \times d_1$ matrices V_{in} and V_{out}. Such a directed model would further complicate identifiability and interpretation. There is now a scale unidentifiability for each latent dimension, which means that we cannot distinguish between the contributions of incoming and outgoing node behaviour.

Finally, recent work has demonstrated that linear embeddings, which assume a low-rank structure on expected adjacency matrices, may be too restrictive for modelling complex real data (Rubin-Delanchy, 2020). In our data application, we find our latent embedding of worldwide agricultural trade to have a relatively high dimension compared with the number of nodes. There may be potential to further reduce the latent dimension by applying additional manifold dimension reduction to the common and individual embeddings.

Acknowledgement

This research was partially supported by the U.S. National Science Foundation.

Supplementary material

Supplementary Material available at *Biometrika* online includes a more detailed description of the proximal gradient descent algorithm in § 3.2, technical proofs of the mathematical results, additional results on synthetic networks, and further analysis of the trade network data in § 6.

References

Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C. E. & Vogelstein, J. T. (2021). Inference for multiple heterogeneous networks with a common invariant subspace. *J. Mach. Learn. Res.* **22**, 1–49.

Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V. & Qin, Y. (2017). Statistical inference on random dot product graphs: A survey. *J. Mach. Learn. Res.* **18**, 8393–484.

Bandeira, A. S. & Van Handel, R. (2016). Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *Ann. Prob.* **44**, 2479–506.

Bickel, P., Choi, D., Chang, X. & Zhang, H. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *Ann. Statist.* **41**, 1922–43.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *Ann. Statist.* **43**, 177–214.

D'angelo, S., Murphy, T. B. & Alfò, M. (2019). Latent space modelling of multidimensional networks with application to the exchange of votes in Eurovision song contest. *Ann. Appl. Statist.* **13**, 900–30.

De Domenico, M., Nicosia, V., Arenas, A. & Latora, V. (2015). Structural reducibility of multilayer networks. *Nature Commun.* **6**, 1–9.

De Vito, R., Bellio, R., Trippa, L. & Parmigiani, G. (2019). Multi-study factor analysis. *Biometrics* **75**, 337–46.

Fithian, W. & Mazumder, R. (2018). Flexible low-rank statistical modeling with missing data and side information. *Statist. Sci.* **33**, 238–60.

Gavish, M. & Donoho, D. L. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans. Info. Theory* **60**, 5040–53.

Gollini, I. & Murphy, T. B. (2016). Joint modeling of multiple network views. *J. Comp. Graph. Statist.* **25**, 246–65.

Handcock, M. S., Raftery, A. E. & Tantrum, J. M. (2007). Model-based clustering for social networks. *J. R. Statist. Soc.* A **170**, 301–54.

Hoff, P. D., Raftery, A. E. & Handcock, M. S. (2002). Latent space approaches to social network analysis. *J. Am. Statist. Assoc.* **97**, 1090–8.

Holland, P. W., Laskey, K. B. & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks* **5**, 109–37.

Jones, A. & Rubin-Delanchy, P. (2021). The multilayer random dot product graph. *arXiv:* 2007.10455v3.

Kim, B., Lee, K. H., Xue, L. & Niu, X. (2018). A review of dynamic network models with latent variables. *Statist. Surv.* **12**, 105–35.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y. & Porter, M. A. (2014). Multilayer networks. *J. Complex Networks* **2**, 203–71.

Lei, J. & Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *Ann. Statist.* **43**, 215–37.

Levin, K., Athreya, A., Tang, M., Lyzinski, V. & Priebe, C. E. (2017). A central limit theorem for an omnibus embedding of multiple random dot product graphs. In *2017 IEEE Int. Conf. Data Mining Workshops (ICDMW)*. New York: Institute of Electrical and Electronics Engineers, pp. 964–7.

Li, T., Levina, E. & Zhu, J. (2020). Network cross-validation by edge sampling. *Biometrika* **107**, 257–76.

Lock, E. F., Park, J. Y. & Hoadley, K. A. (2020). Bidimensional linked matrix factorization for pan-omics pan-cancer analysis. *arXiv:* 2002.02601.

Ma, Z., Ma, Z. & Yuan, H. (2020). Universal latent space model fitting for large networks with edge covariates. *J. Mach. Learn. Res.* **21**, 1–67.

Matias, C. & Robin, S. (2014). Modeling heterogeneity in random graphs through latent space models: A selective review. *ESAIM Proc. Surv.* **47**, 55–74.

Mazumder, R., Hastie, T. & Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.* **11**, 2287–322.

Nielsen, A. M. & Witten, D. (2018). The multiple random dot product graph model. *arXiv:* 1811.12172.

R Development Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. http://www.R-project.org.

Rubin-Delanchy, P. (2020). Manifold structure in graph embeddings. In *Proc. 34th Conf. Neural Information Processing Systems (NeurIPS 2020)*. NeurIPS Proceedings.

Rubin-Delanchy, P., Priebe, C. E., Tang, M. & Cape, J. (2020). A statistical interpretation of spectral embedding: The generalised random dot product graph. *arXiv:* 1709.05506v4.

Salter-Townshend, M. & McCormick, T. H. (2017). Latent space models for multiview network data. *Ann. Appl. Statist.* **11**, 1217–44.

Sosa, J. & Betancourt, B. (2021). A latent space model for multilayer network data. *arXiv:* 2102.09560.

Wang, L., Zhang, Z. & Dunson, D. (2019). Common and individual structure of brain networks. *Ann. Appl. Statist.* **13**, 85–112.

Wang, S., Arroyo, J., Vogelstein, J. T. & Priebe, C. E. (2021). Joint embedding of graphs. *IEEE Trans. Pat. Anal. Mach. Intel.* **43**, 1324–36.

Wu, Y.-J., Levina, E. & Zhu, J. (2017). Generalized linear models with low rank effects for network data. *arXiv:* 1705.06772.

Young, S. J. & Scheinerman, E. R. (2007). Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*. Berlin: Springer, pp. 138–49.

Zhang, X., Xue, S. & Zhu, J. (2020). A flexible latent space model for multilayer networks. *Proc. Mach. Learn. Res.* **119**, 11288–97.

[*Received on* 30 *December* 2020. *Editorial decision on* 18 *October* 2021]