Computing WSBM marginals with Tensor-Train decomposition

Mohamed Anwar Abouabdallah, Olivier Coulaud, Nathalie Peyrard, Alain Franc

Computing WSBM marginals with Tensor-Train decomposition

Mohamed Anwar Abouabdallah[1,2], Olivier Coulaud[3], Nathalie Peyrard[4*], Alain Franc[1,2]

[1]Université de Bordeaux, INRAE, UMR BIOGECO, Cestas, 33612, France.
[2]Pleiade, EPC INRIA-INRAE, Université de Bordeaux, Talence, 33405, France.
[3]Concace, EPII INRIA, Talence, 33405, France.
[4*]Université de Toulouse, INRAE, UR MIAT, Castanet-Tolosan, 31320, France.

*Corresponding author(s). E-mail(s): nathalie.peyrard@inrae.fr;
Contributing authors: maabouabdallah@gmail.com;
olivier.coulaud@inria.fr; alain.franc@inrae.fr;

Abstract

The Weighted Stochastic Block Model (WSBM) is a statistical model for unsupervised clustering of individuals based on a pairwise distance matrix. The probabilities of group membership are computed as unary marginals of the joint conditional distribution of the WSBM, whose exact evaluation with brute force is out of reach beyond a few individuals. We propose to build an exact Tensor-Train (TT) decomposition of the multivariate joint distribution, from the SVD of each binary factor of a WSBM, which leads to variables separation. We present how to exploit this decomposition to compute unary and binary marginals. They are expressed without approximation as products of matrices involved in the TT decomposition. However, the implementation of the procedure faces several numerical challenges. First, the dimensions of the matrices involved grow faster than exponentially with the number of variables. We bypass this difficulty by using the format of TT-matrices. Second, the TT-rank of the products grows exponentially. Then, we use a numerical approximation of matrices product that guarantees a low TT-rank, the rounding. We compare the TT approach with two classical inference methods, the Mean-Field approximation and the Gibbs Sampler, on the problem of binary marginal inference for WSBM with

various distances structures and up to fifty variables. The results lead to recommend the TT approach for its accuracy and reasonable computing time. Further researches should be devoted to the numerical difficulties for controlling the rank in rounding, to be able to deal with larger problems.

Keywords: Binary marginals, Weighted Stochastic Block Model, Variables separation, Tensor-Train format, low rank approximation, TT matrices.

1 Introduction

Pattern discovery in data sets has become a challenging goal in a large diversity of application domains. It is not so easy to define what an interesting pattern is, but among the quasi infinite number of possibilities, manifold learning for continuous patterns and clustering for discrete patterns are two mainstays for building a compact representation of a data set. Here, we are interested in cluster discovery formalised by a statistical model because it quantifies the uncertainties as limits to our knowledge through probabilities for group membership instead of yes/no answer. In many situations, one knows pairwise distances or dissimilarities between a set of items (see Deza and Deza, 2016). An example is distances between DNA sequences in an environmental sample, which are distances on strings computed from local alignment score (Gusfield, 1997; Smith and Waterman, 1981). The question of clustering items based on their pairwise distances is also classical in time series clustering (Ansari et al., 2020). For distance-based clustering, Weighted Stochastic Block Models (WSBM) are more and more used with a large scope of application domains like social sciences (Barbillon et al., 2017), analysis of ecological interaction networks (Miele and Matias, 2017), neurology (Faskowitz et al., 2018), or numerical taxonomy (Abouabdallah et al., 2022). This statistical model formalises the groups the items belong to as latent variables. One of their main advantages is the possibility to find out groups which are not communities (the items within a groupare not necessarily close to each others), like dissortative structures which exist in bipartite graphs. This property gives them great flexibility to adapt to a variety of structures and widen the scope of patterns which can be discovered beyond community detection.

Tools exist to estimate the parameters of a WSBM, as a prerequisite before inferring the group memberships. The classical approach for parameters estimation in models with latent variables is the EM algorithm (Dempster et al., 1977) which, at step E, requires the computation of unary and binary marginals of the conditional distribution of the group memberships. If there are n items and Q groups, computing a unary WSBM marginal requires Q^{n-1} sums and it is a known bottleneck. The main approach used to overpass this difficulty is the Variational EM, where an assumption of independence is made and unary marginals are approximated as fixed points of the Mean-Field equations. (Daudin et al., 2008; Mariadassou et al., 2010). Beyond the utility of marginalisation for parameters estimation, binary marginals of a WSBM are useful for inferring group memberships as their provide a richer information than the

sole unary marginals. Indeed, a group allocation based solely on the unary marginals, i.e. by assigning the individual to the group with higher probability, may be associated with a large uncertainty when the mode is not peaked. In this case, the binary marginal can inform on the source of uncertainty. Indeed, focusing on a particular individual in group g, one can compute the probabilities for it to be in the same group than each other individual of group g. They will reveal if there is equal uncertainty of co-membership or on contrary if uncertainty is due to few specific members.

Therefore, in this work, we consider the problem of computing binary marginals in a WSBM. We will use the fact that a WSBM is a graphical model. A graphical model is a stochastic model in which the joint distribution of the variables can be expressed as a product of factors, each factor involving only a subset of the variables. In a WSBM, the joint distribution of interest is the conditional distribution of the group memberships, and it is a pairwise graphical model (factors are binary at most). For some graphical models, the factorisation property makes marginalisation simple. It depends on the structure of the graph associated to the graphical model, a topic which has been thoroughly studied (see e.g. Wainwright and Jordan, 2008; Koller and Friedman, 2009; Murphy, 2012; Peyrard et al., 2019). Very often, one needs not only to compute one marginal for one variable, but all marginals for all variables, or subsets of variables. This global calculation is performed exactly when the graph of the model is a tree, with message passing or belief propagation. On graphs with loops it can sometimes be performed exactly with message passing using a Junction Tree built from the graph, and otherwise the marginals are approximated as fixed points of loopy belief propagation. The tree width, a feature of the graph associated to the graphical model, characterises the complexity of the calculation of the marginalisation. If the tree-width is w, this complexity is in $\mathcal{O}(kQ^{w+1})$ if there are Q possible values for a discrete random variable. In case of large tree-width, the exact calculation by message passing in the Junction-Tree becomes impossible.

The graph associated to the conditional distribution of the group memberships in a WSBM is a clique. The Junction Tree has one node (the clique), and the tree-width is $n - 1$. It is the most complex situation with n nodes. Therefore, the issue of marginalisation in WSBM should be addressed with other approaches. The above-mentioned Mean-Field approach is a classical option, and MCMC techniques (Robert and Casella, 2004), like the Gibbs sampler is another. The first one is quick but is limited to unary marginals, and the second allows calculation of binary marginals but to the price of a higher computational cost.

In this work we explore an alternative to the Mean-Field and the MCMC approaches for approximate marginals inference in a WSBM. We use the fact that the joint distribution of a graphical model with n variables and Q possible values for each random variable is a tensor of order n and dimension Q per mode (see section 3 for the definition of these terms). If the joint distribution of a set of random variables can be expressed as a tensor of low rank, then marginalisation becomes fairly easy. One possible strategy for marginalisation of the joint distribution is to find the best

approximation in the sense of Frobenius norm of this distribution by a tensor of low rank, and to perform marginalisation on the low rank approximation. There exists several notions of rank for tensors (see e.g. Coppi and Bolasco, 1989; Kolda and Bader, 2009), among which CP-rank (Harshman, 1970; Carroll and Chang, 1970), Tucker rank (Tucker, 1966; Kroonenberg and de Leeuw, 1980; de Lathauwer et al., 2000), and TT-rank (Oseledets, 2009, 2011; Al Daas et al., 2022). It is now acknowledged that TT decomposition is equivalent to the notion of Matrix Product States which has been widely developed in statistical physics (Fannes et al., 1992; Orus, 2014).

Approximation of tensors which correspond to the Conditional Probability Tables (CPT) of a Bayesian Network (BN, directed graphical models) by decompositions based on CP-rank ones tensors has been exploited in Savicky and Vomlel (2007). The authors establish theoretical results on the closed-form expression of the minimal rank-one decomposition for some particular forms of CPT and propose a numerical method to compute a rank-one decomposition for the other situations. In the same spirit, Wrigley et al. (2017) propose an approximation of the factors of a graphical model by rank-one tensors decompositions to implement the Junction Tree algorithm. Marginalisation becomes easy because approximated factors are fully factored (i.e. product of functions of scope of size 1). Multiplication is not as easy and they propose to sample to approximate the product of two factors. However, finding the best low CP-rank approximation of a given tensor is not an easy task. It has been shown that such a problem may be ill-posed because the manifold of tensors of a given CP-rank may not be closed (de Silva and Lim, 2008). Close to its boundary, numerical algorithm may not converge.

The Tucker decomposition is exploited in Lyu et al. (2023). The authors define a new family of graphical models, the latent space models, to study the organisation of individuals linked by hyperedges. To estimate the model parameters and restore the latent positions of the individuals, they propose to maximise the probability of the observed adjacency tensor knowing the latent variables, under a constraint on the Tucker ranks of the tensor of interest. The approach does not requires marginal inference, and the authors do not consider this task. Moreover marginalisation with the Tucker decomposition remains computationally complex.

On the contrary, finding the best TT rank r approximation of a given tensor is a well-posed problem, numerical algorithms for computing it are robust (Oseledets, 2009, 2011), and as we will see the TT format is well adapted for marginalisation because it leads to variables separation. Ducamp et al. (2020) use the TT format of the factors of a BN to write and implement the equations of the classical message passing algorithm for probabilistic inference (i.e. to compute some marginals of the BN). They show on several BN examples how this approach can lead to a reduction in the storage and a faster inference time compared to the exact message passing algorithm, with a reasonable error on the marginals values. This property has been exploited as well in Novikov et al. (2014) to compute the normalising constant and

the unary marginals of any graphical model by first finding a best low rank TT-approximation of each factor and, second, writing the product of the approximations of the factors as a tensor in TT-format by a clever utilisation of the mixed-product property of the Kronecker product.

Our work aims at proposing a way to compute the binary marginals of a WSBM, and more generally of a graphical model with binary factors, by using the TT format of the tensor associated to the joint distribution of the model. The article is organised as follows. We first recall some definitions and notations that we will use throughout this paper, on WSBM in section 2, and on tensors in section 3. The material in those sections is standard. In section 4, we present in detail the approach of Novikov et al. (2014). Our contribution is based on an extension of their work, and is developed in the next 3 sections. In section 5, we observe that the Singular Value Decomposition (SVD) of the matrices which are tables of binary factors is a variable separation between indices of rows and of columns. When plugged in step 1 in Novikov procedure, this leads to writing the joint distribution of a pairwise graphical model exactly as a tensor in TT format, without approximation. Distributivity of multiplication over addition in the ring of matrices leads to an algebraically simple calculation of the maginals. We develop a approach by recursion for the simultaneous calculation of all binary marginals, over all variables and all possible values the variables, which extends Novikov's procedure for computing all unary marginals. If such an approach is algebraically straightforward, it leads to sums and products between matrices with huge dimensions. In section 6, elaborating on the TT-matrix approach used in Novikov et al. (2014), we show that it leads to numerical locks for which some solutions are proposed, essentially based on the rounding (Oseledets, 2011; Al Daas et al., 2022). Finally, in section 7, we compare both in time and accuracy the computation of binary marginals by our TT approach to three classical methods: brute force enumeration, simulation with a Gibbs sampler, and as fixed point of a Mean-Field approximation.

2 Weighted Stochastic Block Model

The Stochastic Block Model (SBM, Holland et al., 1983; Daudin et al., 2008) was originally defined for identifying groups in a set of individuals connected by binary relationships (two individuals are either connected or not). In this work, we consider the extension to dissimilarity matrices, where to each pair of individuals is attached a weight which is a measure of dissimilarity between the two individuals.

The Weighted SBM (WSBM) is a model from the family of statistical models with latent variables. Let us consider n individuals. The observed variable is the dissimilarity matrix D of size $n \times n$, with element $D[i, j]$. The latent (unobserved) variables are the group memberships of each individual: $Z_i \in \{1, \dots, Q\}$ is the group of individual i. The model relies on two assumptions. First, the Z_i's are independent and their distribution is described by the vector of probabilities $\alpha = (\alpha_1, \dots, \alpha_Q)$, such that $P(Z_i = q) = \alpha_q$. Second, the distribution of the dissimilarity between i and j depends only on the groups of i and j (an not on the individuals i and j). We assume that

the distribution of $D[i,j]$ conditionally to (Z_i, Z_j) depends on a parameter Λ and we denote $\theta = (\alpha, \Lambda)$ the model's parameters. For instance, $\mathbb{P}(D[i,j] \mid Z_i = q, Z_j = q')$ can be the Poisson distribution with parameter $\lambda_{qq'}$. The connectivity matrix of the model is the Q by Q matrix Λ such that $\Lambda[q,q'] = \lambda_{qq'}$. Let $Z = (Z_1, \ldots, Z_n)$. Then, the joint probability distribution of Z and D is:

$$\begin{aligned}\mathbb{P}_\theta(Z, D) &= \mathbb{P}_\Lambda(D \mid Z)\mathbb{P}_\alpha(Z) \\ &= \left(\prod_{i=1}^{n}\prod_{i>j}\mathbb{P}_\Lambda(D[i,j] \mid Z_i, Z_j)\right)\prod_{i=1}^{n}\mathbb{P}_\alpha(Z_i) \end{aligned} \tag{1}$$

In practice, when modelling a dissimilarity matrix with a WSBM the main goal is to recover each individual's memberships, i.e. to restore the Z_is. This is performed by maximising the conditional distribution $\mathbb{P}_\theta(Z \mid D)$. Beyond this 'definitive' affectation of an individual to a group, one may be interested by the vector of probabilities that an individual belongs to each group, namely $\{\mathbb{P}_\theta(Z_i = q \mid D)\}_{1\leq q\leq Q}$, to quantify the uncertainty on the membership. Another probability of interest is the joint conditional probability that two individuals i and j belong to the same group: $\mathbb{P}_\theta(Z_i = q, Z_j = q \mid D)$. Therefore, the distributions of interest are not only the conditional distribution $\mathbb{P}_\theta(Z \mid D)$ but also its unary and binary marginals. The probability $\mathbb{P}_\theta(Z \mid D)$ is proportional to $\mathbb{P}_\theta(Z, D)$,

$$\mathbb{P}_\theta(Z \mid D) = \mathbb{P}_\theta(Z, D)/W \tag{2}$$

and the normalising constant W is equal to the likelihood of the observed dissimilarity matrix D:

$$\begin{aligned} W &= \sum_{z\in\{1,\ldots,Q\}^n}\mathbb{P}_\theta(Z = z, D) \\ &= \mathbb{P}_\theta(D) \end{aligned}$$

The binary marginal for two individuals, for instance 1 and 2, is defined by

$$\begin{aligned}\mathbb{P}_\theta(Z_1 = q, Z_2 = q' \mid D) &= \sum_{z_3=1}^{Q}\cdots\sum_{z_n=1}^{Q}\mathbb{P}_\theta(Z_1 = q, Z_2 = q', Z_3 = z_3, \ldots, Z_n = z_n \mid D) \\ &= \frac{1}{W}\sum_{z_3=1}^{Q}\cdots\sum_{z_n=1}^{Q}\mathbb{P}_\theta(Z_1 = q, Z_2 = q', Z_3 = z_3, \ldots, Z_n = z_n, D)\end{aligned} \tag{3}$$

The unary marginal $\mathbb{P}_\theta(Z_i = q \mid D)$ is obtained by marginalising any binary marginal involving Z_i:

$$\mathbb{P}_\theta(Z_i = q \mid D) = \sum_{q'=1}^{Q}\mathbb{P}_\theta(Z_i = q, Z_j = q' \mid D)$$

In practice, un-normalised binary marginals, $\tilde{P}_\theta(Z_i = q, Z_j = q' \mid D) = \mathbb{P}_\theta(Z_i = q, Z_j = z_j, D)$ are computed, by marginalising $\mathbb{P}_\theta(Z, D)$ instead of $\mathbb{P}_\theta(Z \mid D)$. Then, un-normalised unary marginals $\tilde{P}_\theta(Z_i = q \mid D) = \mathbb{P}_\theta(Z_i = q, D)$ are derived by marginalisation of the $\tilde{P}_\theta(Z_i = q, Z_j = q' \mid D)$. Finally W is obtained from any un-normalised unary marginal as $W = \sum_{q=1}^{Q} \tilde{P}_\theta(Z_i = q \mid D)$.

For the following, it will be convenient to cast the WSBM in the family of graphical models (Koller and Friedman, 2009). A vector of random variables $Z = \{Z_1, \ldots, Z_n\}$ is a graphical model if the joint distribution can be expressed (up to the normalising constant) as a product of functions, called factors, involving only subsets of the variables. From expression (1), we can see that in a WSBM, the distribution $\mathbb{P}_\theta(Z \mid D)$ is that of a graphical model with only binary and unary factors. There are $\frac{n(n-1)}{2}$ binary factors, equal to $\mathbb{P}_\Lambda(D[i,j] \mid Z_i, Z_j)$ and n unary factors equal to $\mathbb{P}_\alpha(Z_i)$. If computed by applying naïvely expression (3), the complexity of evaluation of one binary marginal is in $\mathcal{O}(Q^{n-2})$.

The Mean-Field approximation of a WSBM is its best approximation, in the sense of the Kullback-Leibler divergence, by a product of independent unary factors ϕ.

$$\mathbb{P}_\theta(Z_1 = z_1, \ldots, Z_n = z_n, D) \approx \prod_{i=1}^{n} \phi_i(z_i)$$

In such a case, we have separation of variables, and marginalisation is simple due to the distributivity of multiplication over addition:

$$\sum_{z_1=1}^{Q} \cdots \sum_{z_n=1}^{Q} \left(\prod_{i=1}^{n} \phi_i(z_i) \right) = \prod_{i=1}^{n} \left(\sum_{z_i=1}^{Q} \phi_i(z_i) \right) \tag{4}$$

which requires nQ sums and $n-1$ products. Such a distributivity exists as well in the ring of matrices. Therefore, marginalisation with separation of variables can be done as well in the case where the joint distribution of the n variables can be approximated by a product of unary factors G, each being a matrix, like, for $n = 3$

$$\mathbb{P}_\theta(Z_1 = z_1, \ldots, Z_n = z_n, D) \approx G_1(z_1)\, G_2(z_2)\, G_3(z_3) \qquad \text{with} \quad \begin{cases} G_1(z_1) & \in \mathbb{R}^{1 \times r_1} \\ G_2(z_2) & \in \mathbb{R}^{r_1 \times r_2} \\ G_3(z_3) & \in \mathbb{R}^{r_2 \times 1} \end{cases} \tag{5}$$

Such a decomposition is known as Tensor-Train decomposition (TT), and has been thoroughly studied in the literature (see next section). We will see in Section 5 that the joint distribution of a WSBM (or any pairwise graphical model) can be exactly decomposed in a TT format, which paves the way for its marginalisation. Indeed, in

the simple case of the probability written in TT-format in equation (5), we have

$$\begin{aligned} W &= \sum_{z_1=1}^{Q} \sum_{z_2=1}^{Q} \sum_{z_3=1}^{Q} G_1(z_1)\, G_2(z_2)\, G_3(z_3) \\ &= \left(\sum_{z_1=1}^{Q} G_1(z_1) \right) \left(\sum_{z_2=1}^{Q} G_2(z_2) \right) \left(\sum_{z_3=1}^{Q} G_3(z_3) \right) \end{aligned} \tag{6}$$

3 Tensors and Tensor Trains

The main idea which has driven this study is to consider that the joint distribution of a WSBM defines a tensor, i.e. a multi-way array. Then, following Novikov et al. (2014), we use possibilities given by computing in TT-format (mainly due to Oseledets, 2009, 2011) to separate the variables for each variable z_i of the WSBM in the expression of $\mathbb{P}_\theta(Z, D)$. This separation enables to compute the marginals of the joint distribution. Before positioning our method, we recall here the main definitions and results on tensors we will use in the rest of the article.

Tensors:

Let $\mathbf{E} = (E_1, \ldots, E_i, \ldots, E_n)$ be a family of n finite dimensional real vector spaces. A tensor $\mathbf{T}$ on $\mathbf{E}$ is a multilinear form on $E_1 \times \ldots \times E_n$. Then, n is the order of the tensor, E_i for $1 \leq i \leq n$ is a mode of $\mathbf{T}$ (often simplified as i) and the dimension of E_i, denoted Q_i, is its dimension for mode E_i. The multilinear dimension of the tensor is denoted $Q_1 \times \ldots \times Q_n$, with $\mathbf{T} \in \mathbb{R}^{Q_1 \times \ldots \times Q_n}$. If a basis has been selected on each space E_i, $\mathbf{T}$ can be represented in those basis as a $n-$dimensional array, the elements of which are denoted

$$\mathbf{T}[z_1, \ldots, z_i, \ldots, z_n] \qquad \text{with} \quad 1 \leq i \leq n, \quad 1 \leq z_i \leq Q_i,$$

or $\mathbf{T}_{z_1 \ldots z_n}$.

Slice of a tensor:

Let us explain what a slice of a tensor is on the example of a $3-$modes tensors on $E_1 \times E_2 \times E_3$ with indices z_1, z_2, z_3. A general term of $\mathbf{T}$ is $\mathbf{T}[z_1, z_2, z_3]$ and $\mathbf{T}$ has dimension $Q_1 \times Q_2 \times Q_3$ with $1 \leq z_1 \leq Q_1$, $1 \leq z_2 \leq Q_2$ and $1 \leq z_3 \leq Q_3$. The slice $\mathbf{T}_{\mathrm{E}_2}(z_2)$ on mode E_2 with index z_2 or, more simply, $\mathbf{T}_2(z_2)$, is the tensor of dimension $Q_1 \times Q_3$ obtained by letting the indices in modes E_1 and E_3 ($\neq E_2$) run over all dimensions of respectively E_1 and E_3 and fixing the index for mode E_2 at z_2:

$$\mathbf{T}_2(z_2) = \left(\mathbf{T}[z_1, z_2, z_3]\right)_{z_1, z_3} \qquad \text{with} \quad 1 \leq z_1 \leq Q_1, \quad 1 \leq z_3 \leq Q_3.$$

or, more simply:

$$\mathbf{T}_2(z_2) = \mathbf{T}[:, z_2, :]$$

This can be extended with some technicalities in notations to any tensor, by selecting a mode E_i (referred to as $\mathbf{T}_i$) and an index z_i ($1 \leq z_i \leq Q_i$):

$$\mathbf{T}_i(z_i) = \mathbf{T}[:, \ldots, :, z_i, :, \ldots, :]$$

Frobenius norm of a tensor:

The set of tensors of a gi ven multilinear dimension is a vector space, which can inherit the Euclidean structure of all modes. The Frobenius norm $\|\mathbf{T}\|$ of $\mathbf{T}$ is defined by

$$\|\mathbf{T}\|^2 = \sum_{i_1=1}^{Q_1} \ldots \sum_{i_n=1}^{Q_n} \mathbf{T}^2[i_1 \ldots i_n].$$

This will be useful to define a distance between tensors, as $d(\mathbf{T}, \mathbf{T}') = \|\mathbf{T} - \mathbf{T}'\|$.

Tensor-Train format:

The Tensor-Train format, denoted TT-format in the rest of this article, is a format proposed in Oseledets (2009, 2011), well adapted to separation of variables (it can be read as a local, or componentwise, separation of variables). A tensor $\mathbf{T} = (\mathbf{T}[z_1, \ldots, z_n])_{z_1,\ldots,z_n}$ is in TT-format if there exists matrices $G_i(z_i)$ for $1 \leq i \leq n$ such that

$$\forall\, (z_1, \ldots, z_n), \quad \text{with } 1 \leq z_i \leq Q_i, \quad \mathbf{T}[z_1, \ldots, z_n] = G_1(z_1) \ldots G_n(z_n),$$

(Oseledets, 2011, equation (1.2)). The matrices $G_i(z_i)$ are called the *cores* of the TT-decomposition of $\mathbf{T}$. This is explicitly variable separation for the modes of the tensor. The dimensions of matrix $G_i(z_i)$ are $r_{i-1} \times r_i$, with $r_0 = r_n = 1$. These are called the ranks of the TT-format. The TT-rank of $\mathbf{T}$ is

$$\mathbf{r} = (r_0, \ldots, r_n).$$

The TT-decomposition of $\mathbf{T}$ can be developed by defining componentwise for each mode i

$$G_i(z_i) = (G_i(z_i)[\alpha_{i-1}, \alpha_i])_{\alpha_{i-1},\alpha_i} \in \mathbb{R}^{r_{i-1} \times r_i} \qquad \text{with} \quad \begin{cases} 1 \leq \alpha_{i-1} \leq r_{i-1} \\ 1 \leq \alpha_i \leq r_i \end{cases}$$

and the tensor $\mathbf{G}_i$ of order 3 of dimension $r_{i-1} \times Q_i \times r_i$ with coefficients $\mathbf{G}_i[\alpha_{i-1}, z_i, \alpha_i]$. The matrix $G_i(z_i)$ is the slice z_i of second mode of tensor $\mathbf{G}_i$. This leads to

$$\forall (z_1, \ldots, z_n), \quad \mathbf{T}[z_1, \ldots, z_n] = \sum_{\alpha_1=1}^{r_1} \sum_{\alpha_2=1}^{r_2} \ldots \sum_{\alpha_{n-1}=1}^{r_{n-1}} \mathbf{G}_1[z_1, \alpha_1] \mathbf{G}_2[\alpha_1, z_2, \alpha_2] \ldots \mathbf{G}_n[\alpha_{n-1}, z_n]$$

Vidal (2003) has shown in the framework of the MPS that such a decomposition always exists, and is not unique. However, a simple dimension analysis shows that the

rank may be huge ($r \sim \sqrt{Q^{n-1}/n}$) in general). A classical problem is, a tensor $\mathbf{T}$ and a rank $\mathbf{r}$ being given, to find a tensor in TT format of rank $\mathbf{r}$ which is closest to $\mathbf{T}$ with Frobenius norm. Oseledets (2011) has proposed a robust algorithm, called TT-SVD, to solve this optimisation problem (see Oseledets, 2011, Algorithm 1). He has shown that many basic operations in tensor calculus can be done in the framework of the TT-format, i.e. knowing the cores only, without using explicitly the coefficients of the tensors involved: addition, contraction, Hadamard product, inner product, matrix-by-vector product, and the best approximation at a given TT-rank called the rounding. Bypassing the explicitation of all coefficients leads to powerful tools for many modes tensors which cannot be stored in memory.

Suitability for marginalisation:

Marginalisation in tensors is an operation associated with slicing. It consists in (i) selecting a slice, which can be empty and (ii) summing up all terms which are not in the slice. Here, we show why TT format is suitable for marginalisation. Let us first develop it for a tensor of order 3 with equal dimension Q on all modes, as

$$\forall 1 \leq z_1, z_2, z_3 \leq Q, \quad \mathbf{T}[z_1, z_2, z_3] = G_1(z_1)\, G_2(z_2)\, G_3(z_3) \qquad \text{with} \quad \begin{cases} G_1(z_1) & \in \mathbb{R}^{1 \times r_1} \\ G_2(z_2) & \in \mathbb{R}^{r_1 \times r_2} \\ G_3(z_3) & \in \mathbb{R}^{r_2 \times 1} \end{cases}$$

Let us consider the slice defined by mode E_2 and index z_2 to define the marginal $m_2(z_2)$. Then, because of distributivity of multiplication on addition in the ring of matrices

$$\begin{aligned} m_2(z_2) &= \sum_{z_1, z_3 = 1}^{Q} \mathbf{T}[z_1, z_2, z_3] \\ &= \sum_{z_1, z_3 = 1}^{Q} G_1(z_1) G_2(z_2) G_3(z_3) \\ &= \left(\sum_{z_1=1}^{Q} G_1(z_1) \right) G_2(z_2) \left(\sum_{z_3=1}^{Q} G_3(z_3) \right) \\ &= B_1\, G_2(z_2)\, B_3 \end{aligned} \tag{7}$$

with

$$B_1 = \sum_{z_1=1}^{Q} G_1(z_1) \in \mathbb{R}^{1 \times r_1}, \qquad B_3 = \sum_{z_3=1}^{Q} G_3(z_3) \in \mathbb{R}^{r_2 \times 1}.$$

This can be extended in a straightforward way to tensors of any order n with unequal dimensions.

4 TT to compute the normalising constant of a graphical model

In this section we present in detail the approach developed in Novikov et al. (2014). They use the TT-approximation of the factors of a graphical model to compute the

normalising constant. Our work on inference for WSBM is elaborated on their results. We present in Table 1 a few notations used in the rest of the article that allow a transition between the domains of statistical modelling and tensor algebra.

symbol	WSBM	tensor
n	# of individuals	# of modes
i	an individual	a mode
Q	number of groups	dimension of a mode
z_i	latent group membership for individual i	index in mode i $z_i \in \{1, \dots, Q\}$
$\psi_{ij}(z_i, z_j)$	a factor in WSBM	
$\Psi_{ij}[z_i, z_j]$		matrix of the table associated to ψ_{ij}
$\psi(z_1, \dots, z_n)$	joint distribution	
$\boldsymbol{\Psi}[z_1, \dots, z_n]$		associated tensor
m	# of factors $n(n-1)/2$	–
ℓ	index of a factor	–

Table 1 Notations for transition between WSBM and tensor calculus.

When a is a vector or M a matrix, we denote by $a[i]$ the component i of a and by $M[i, j]$ the component at row i and column j of M. If, for example, ψ_{ij} is a binary factor of a graphical model on edge (i, j), represented by a matrix Ψ_{ij}, if the group membership of individual i (resp. j) is z_i (resp. z_j), both in state space S, we denote by $\psi_{ij}(z_i, z_j)$ the factor as a function from $S \times S$ on $\mathbb{R}$ and by $\Psi_{ij}[z_i, z_j]$ the coefficients of the matrix associated to it.

Let us have a graphical model with n variables, each representing an individual, with z_i being the group of individual i. The graphical model is defined on a hypergraph $G = (I, E)$ where $I = \{1, \dots, n\}$ and E is a set of hyperedges $A_\ell \subset I$ for $\ell \in \{1, \dots, m\}$. Let us denote $n_\ell = |A_\ell|$, and by $\mathbf{z}_\ell$ the groups of the individuals in A_ℓ i.e. $\mathbf{z}_\ell = (z_i)_{i \in A_\ell}$. Then, the un-normalised distribution of the graphical model can be written as a product of m factors indexed by ℓ:

$$\psi(z_1, \dots, z_n) = \prod_{\ell=1}^{m} \psi_\ell(\mathbf{z}_\ell)$$

The objective of the work in Novikov et al. (2014) is to compute the normalising constant

$$W = \sum_{z_1=1}^{Q} \dots \sum_{z_n=1}^{Q} \left(\prod_{\ell=1}^{m} \psi_\ell(\mathbf{z}_\ell) \right)$$

Their method is composed of four steps presented here as we will elaborate on them to develop our method for computing marginals in a WSBM. Let us denote by $\boldsymbol{\Psi} := \boldsymbol{\Psi}[z_1, \dots, z_n] = \psi(z_1, \dots, z_n)$ the tensor which is the distribution table associated to the un-normalised joint distribution ψ.

Step 1: approximating each factor for a given TT-format

A tensor $\mathbf{\Psi}_\ell$ can be built from each factor ψ_ℓ. Its order is n_ℓ, the number of indices in A_ℓ, and its dimensions are Q for each mode. We define

$$\mathbf{\Psi}_\ell[\mathbf{z}_\ell] = \psi_\ell(\mathbf{z}_\ell), \qquad \text{with} \quad \mathbf{\Psi}_\ell \in \mathbb{R}^{Q\times\ldots\times Q}$$

The first step consists in selecting a rank r and computing a best TT-approximation $\widetilde{\mathbf{\Psi}}_\ell$ of $\mathbf{\Psi}_\ell$ at rank r (more rigorously, at rank $\mathbf{r} = (1, r, \ldots, r, 1)$). In the TT-format, each component $\widetilde{\mathbf{\Psi}}_\ell[z_{i_1^\ell}, \ldots, z_{i_{n_\ell}^\ell}]$ can be written as a product of matrices with separation of variables (here, $\mathbf{z}_\ell = (z_{i_1^\ell}, \ldots, z_{i_{n_\ell}^\ell})$ where i_μ^ℓ is $\mu-$th index in A_ℓ):

$$\widetilde{\mathbf{\Psi}}_\ell[z_{i_1^\ell}, \ldots, z_{i_{n_\ell}^\ell}] = G_{i_1^\ell}^\ell(z_{i_1^\ell}) \ldots G_{i_{n_\ell}^\ell}^\ell(z_{i_{n_\ell}^\ell}), \qquad \text{with} \quad \begin{cases} G_{i_1^\ell}^\ell(z_{i_1^\ell}) \in \mathbb{R}^{1\times r} \\ G_{i_\mu^\ell}^\ell(z_{i_\mu^\ell}) \in \mathbb{R}^{r\times r} \qquad i_2^\ell \le i_\mu^\ell \le i_{n_\ell-1}^\ell \\ G_{i_{n_\ell}^\ell}^\ell(z_{i_{n_\ell}^\ell}) \in \mathbb{R}^{r\times 1}. \end{cases}$$

For sake of simplicity, we select a same rank r for each node, but it can be developed with ranks specific to each node. It may happen that, for a given factor and a given rank, the TT-decomposition at rank r is exact. For example, if $n_\ell = 3$, tensor $\mathbf{\Psi}_\ell$ has Q^3 terms, and a TT of order 3 with rank r has $Qr(r+2)$ terms. A simple dimension analysis shows that if $r = Q$, an exact TT-decomposition likely exists. In order to encompass in a same notation exact decomposition and best approximation at a given rank for any factor, we will use the same notation $\mathbf{\Psi}_\ell$ from now on, and drop the $\widetilde{(.)}$. We then start with

$$\mathbf{\Psi}_\ell[z_{i_1^\ell}, \ldots, z_{i_{n_\ell}^\ell}] = G_{i_1^\ell}^\ell(z_{i_1^\ell}) \ldots G_{i_{n_\ell}^\ell}^\ell(z_{i_{n_\ell}^\ell}).$$

Step 2: Adding non essential variable:

This step is simple, but notations can be cumbersome. So, it will be presented first on the small example of Novikov et al. (2014), section 5.1. The idea is to build for each hyperedge A_ℓ a new factor $\overline{\psi_\ell}$ that depends on all variables and not only on $\mathbf{z}_\ell$ but whose value does not actually depend on z_j for $j \notin A_\ell$:

$$\begin{aligned} \overline{\psi}_\ell \ : \ \{1, \ldots, Q\}^n &\longrightarrow \quad \mathbb{R} \\ (z_1, \ldots, z_n) &\longrightarrow \psi_\ell(\mathbf{z}_\ell) \end{aligned}$$

The tensor $\overline{\mathbf{\Psi}}_\ell$ attached to $\overline{\psi}_\ell$ has order n. We have

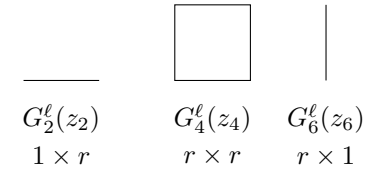
$$\mathbf{\Psi}[z_1, \ldots, z_n] = \prod_{\ell=1}^{m} \overline{\mathbf{\Psi}}_\ell[z_1, \ldots, z_n] \tag{8}$$

The TT-format of tensors $\overline{\mathbf{\Psi}}_\ell$ with non essential variables can be derived from the TT-format of the tensor corresponding to the factor $\overline{\psi}_\ell$ as shown in the following toy

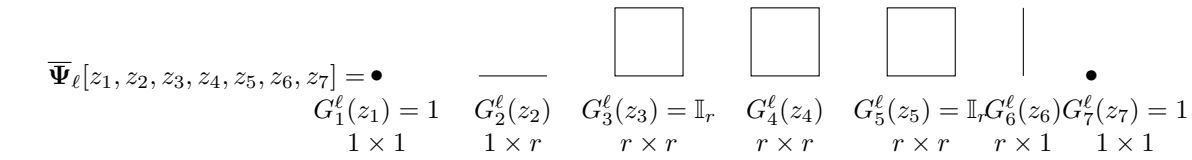
example, with $n = 7$, $A_\ell = \{2, 4, 6\}$. We then have

$$\boldsymbol{\Psi}_\ell[z_2, z_4, z_6] = G_2^\ell(z_2)\, G_4^\ell(z_4)\, G_6^\ell(z_6) \qquad \text{with} \quad \begin{cases} G_2^\ell(z_2) & \in \mathbb{R}^{1\times r} \\ G_4^\ell(z_4) & \in \mathbb{R}^{r\times r} \\ G_2^\ell(z_6) & \in \mathbb{R}^{r\times 1} \end{cases}$$

This can be sketched as

$\boldsymbol{\Psi}_\ell[z_2, z_4, z_6] =$ 

The product of those matrices of dimensions $(1, r)$, (r, r) and $(r, 1)$ is a real. To build the TT-format of $\overline{\boldsymbol{\Psi}}_\ell$, we add new cores, one per non essential variable, which are $G_1^\ell(z_1) = 1$, $G_3^\ell(z_3) = G_5^\ell(z_5) = \mathbb{I}_r$ and $G_7^\ell(z_7) = 1$ as in the following figure:

$\overline{\boldsymbol{\Psi}}_\ell[z_1, z_2, z_3, z_4, z_5, z_6, z_7] =$ 

Finally,

$$\overline{\boldsymbol{\Psi}}_\ell[z_1, z_2, z_3, z_4, z_5, z_6, z_7] = G_1^\ell(z_1)\, G_2^\ell(z_2)\, G_3^\ell(z_3)\, G_4^\ell(z_4)\, G_5^\ell(z_5)\, G_6^\ell(z_6)\, G_7^\ell(z_7)$$

This procedure is applied to each factor of the graphical model. Then, there exists a family of cores $(G_k^\ell(z_k))_{k,\ell}$ with $1 \le \ell \le m$ and $1 \le k \le n$ such that

$$\forall\, A_\ell \in E, \quad \overline{\boldsymbol{\Psi}}_\ell[z_1, \ldots, z_n] = \prod_{k=1}^{n} G_k^\ell(z_k)$$

Step 3: Mixed-product property of Kronecker product:

Let us recall that $\otimes$ denotes the Kronecker product between matrices. Knowing that $xy = x \otimes y$ if $x, y \in \mathbb{R}$, equation (8) can be written

$$\begin{aligned} \boldsymbol{\Psi}[z_1, \ldots, z_n] &= \bigotimes_{\ell=1}^{m} \overline{\boldsymbol{\Psi}}_\ell[z_1, \ldots, z_n] \\ &= \bigotimes_{\ell=1}^{m} \left(\prod_{k=1}^{n} G_k^\ell(z_k) \right) \end{aligned}$$

Let us recall the mixed-product property of Kronecker product (see Horn and Johnson, 2012): if A, B, C, D are matrices with relevant dimensions for the products AB and

CD to be possible, then

$$(AB) \otimes (CD) = (A \otimes C)\,(B \otimes D).$$

This leads to

$$\mathbf{\Psi}[z_1, \ldots, z_n] = \prod_{k=1}^{n} \left(\bigotimes_{\ell=1}^{m} G_k^{\ell}[z_k] \right)$$

Let us denote

$$A_k(z_k) = \bigotimes_{\ell=1}^{m} G_k^{\ell}(z_k) \tag{9}$$

Then, the un-normalised joint distribution ψ can be expressed as

$$\psi(z_1, \ldots, z_n) = \mathbf{\Psi}[z_1, \ldots, z_n] = \prod_{k=1}^{n} A_k(z_k) \tag{10}$$

which is separation of variables.

Step 4: computation of the normalising constant and of unary marginals
It is now easy to compute the normalising constant W of ψ. We recall that

$$W = \sum_{z_1=1}^{Q} \ldots \sum_{z_n=1}^{Q} \psi(z_1, \ldots, z_n)$$

Then,

$$\begin{aligned} W &= \sum_{z_1=1}^{Q} \ldots \sum_{z_n=1}^{Q} \left(\prod_{k=1}^{n} A_k(z_k) \right) \\ &= \prod_{k=1}^{n} \left(\sum_{z_k=1}^{Q} A_k(z_k) \right) \\ &= \prod_{k=1}^{n} B_k, \qquad \text{with} \quad B_k = \sum_{z_k=1}^{Q} A_k(z_k). \end{aligned}$$

It requires nQ matrix additions and $n-1$ matrix multiplications only, if all matrices $A_k(z_k)$ have been computed for all $1 \leq k \leq n$ and $1 \leq z_k \leq Q$ (hence nQ matrices). The complexity is no longer exponential with n, but depends on the dimensions of the matrices A_k.

Expression (10) permits to compute the unary marginals as well, with message passing, as evoked in Novikov et al. (2014). Let us denote $\mathbf{z} \backslash z_i = (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$, i.e. all z_j for $j \neq i$. The un-normalised unary marginal for variable i is defined as

$$m_i(z_i) = \sum_{\mathbf{z} \backslash z_i \in \{1, \ldots, Q\}^{n-1}} \psi(z_1, \ldots, z_n)$$

Then

$$\begin{aligned} m_i(z_i) &= \sum_{\mathbf{z}\setminus z_i} \left(\prod_{k=1}^{n} A_k(z_k) \right) \\ &= \left[\sum_{z_1,\dots z_{i-1}} \left(\prod_{k=1}^{i-1} A_k(z_k) \right) \right] A_i[z_i] \left[\sum_{z_{i+1},\dots z_n} \left(\prod_{k=i+1}^{n} A_k(z_k) \right) \right] \\ &= \left(\prod_{k=1}^{i-1} B_k \right) A_i[z_i] \left(\prod_{k=i+1}^{n} B_k \right) \end{aligned}$$

For one unary marginal, this requires only $n-1$ matrix multiplications. If all marginals for all z_k have to be computed, the calculations can be organised in order to mutualise them. Computing all marginals $m_i(z_i)$ for all i and all z_i requires $2(n-2)+2nQ$ matrix multiplications: $n-2$ to compute recursively each $\prod_{k=1}^{i-1} B_k$ for $i=3$ to n, again $n-2$ to compute recursively each $\prod_{k=i+1}^{n} B_k$ for $i=1$ to $n-2$, and finally $2(n+1)(Q-1)$ (because we use $\sum_q m_i(q)=1$) to compute each $m_i(z_i)$ for all i and all z_i by multiplying three matrices (or only 2 for m_1 and m_n).

This expression does not allow to reveal the real complexity of the calculation, which depends on the dimensions of the matrices. Indeed, if A, B are two $p\times p$ matrices, the complexity of the calculation of the product AB is in $\mathcal{O}(p^3)$. Such dimensions are specific to the graphical model defined by ψ and its factors.

From now on we focus on the case of graphical models with binary factors (as for the WSBM model) and starting from the Novikov approach we establish the following results :

1. We first observe in section 5.1 that if a factor ψ_ℓ is binary, the tensor $\mathbf{\Psi}_\ell$ attached to it is a matrix, and its TT-SVD is the SVD of this matrix
2. For pairwise graphical models (i.e. all factors are binary, like an Ising model), this leads to a rewriting of the tensor $\mathbf{\Psi}$ with separation of variable by equation (10) without any approximation (step 1 is exact). This leads to a calculation of all the binary marginals by a recursion approach inspired by Message Passing (section 5.2)
3. We develop this approach on a specific family of pairwise graphical model: the WSBM, for which all binary factors are present (section 5.3). We show that, even though the calculation is algebraically exact, it can be intractable for large n (due to the matrix $\times$ matrix product of the A_k), and we propose some low rank approximation with the rounding (section 6).

5 Separation of variables and computation of binary marginals for pairwise graphical models

In this section, we show that Novikov approach when implemented on a graphical model with binary factors at most leads to rewrite the un-normalised joint distribution in Tensor-Train format, hence with variable separation for computing normalising constant or marginals. This applies to WSBM models. We first develop the approach

of section 4 on pairwise graphical models, and focus after on the particular case of WSBM, which are pairwise graphical models.

5.1 Separation of variables in the un-normalised joint distribution of a pairwise graphical model

The main advantage of pairwise graphical model is that the TT-format of each binary factor (i.e. separation of variables) is obtained with the SVD of the factor. We implement it as first step in Novikov approach for binary graphical models. Further steps are unchanged, and presented in appendix C.

Let us have a pairwise graphical model with un-normalised distribution ψ defined as follows:

$$\psi(z_1 \dots, z_n) = \prod_{(i,j)\in E} \psi_{ij}(z_i, z_j)$$

where E is the set of binary factors of the graphical model. Factor ψ_{ij} can be given as a $Q \times Q$ matrix Ψ_{ij} with $\Psi_{ij}[z_i, z_j] = \psi_{ij}(z_i, z_j)$ where z_i and z_j label the rows and columns of Ψ_{ij}. The SVD of Ψ_{ij} can be written as (see e.g. Horn and Johnson, 2012; Strang, 2019)

$$\Psi_{ij} = U_{ij}\, \Sigma_{ij}\, V_{ij}^{\mathrm{T}}, \qquad \text{with} \quad U_{ij}, \Sigma_{ij}, V_{ij} \in \mathbb{R}^{Q\times Q}$$

where Σ_{ij} is a diagonal matrix, and U_{ij}, V_{ij} are columnwise orthonormal. We assume that matrices are full rank, i.e. rank $\Psi_{ij} = Q$. Let us denote $M_{ij} = U_{ij}\, \Sigma_{ij}$. Then, $\Psi_{ij} = M_{ij}\, V_{ij}^{\mathrm{T}}$. So

$$\Psi_{ij}[z_i, z_j] = \sum_{q=1}^{Q} M_{ij}[z_i, q]\, V_{ij}[q, z_j]$$

Let us denote by $M_{ij}[a] \in \mathbb{R}^Q$ the row a of M_{ij} and by $V_{ij}[b] \in \mathbb{R}^Q$ the column b of V_{ij}. These can be considered as one row and one column matrices respectively

$$M_{ij}[z_i] \in \mathbb{R}^{1\times Q}, \qquad V_{ij}[z_j] \in \mathbb{R}^{Q\times 1}$$

Then

$$\psi_{ij}(z_i, z_j) = \Psi_{ij}[z_i, z_j] = M_{ij}[z_i]\, V_{ij}[z_j] \qquad \text{(Matrices product)}$$

This is variable separation for $\psi_{ij}(z_i, z_j)$: each matrix depends only on one variable z_i or z_j.

The subsequent steps (adding non essential variables, computation of the matrices $A_k(z_k)$ and B_k) are implemented as in section 4, and are described with a specific implementation for a WSBM in appendix C.

5.2 Computing the binary marginals of a pairwise graphical model

Like the unary marginals, the un-normalised binary marginals can be expressed in terms of the B_k and the $A_k(z_k)$:

$$\begin{aligned} m_{ij}(q,q') &= \tilde{P}_\theta(Z_i = q, Z_j = q' \mid D) \\ &= B_1 \times \ldots \times B_{i-1} \times A_i(q) \times B_{i+1} \times \ldots \times B_{j-1} \times A_j(q') \times B_{j+1} \times \ldots \times B_n \end{aligned}$$

Let us define the following quantities:

$$\begin{aligned} \forall\, 1 \leq i \leq n, \quad & F_{i,i} = B_i \\ \forall\, 1 \leq i \leq n-1, \quad \forall\, i+1 \leq j \leq n, \quad & F_{i,j} = B_i \times B_{i+1} \times \ldots \times B_j \end{aligned}$$

Depending on the values of i and j, $m_{ij}(q,q')$ can be expressed as follows:

$$\begin{array}{|llll} i = 1 & j = n & \Rightarrow m_{1n}(q,q') & = A_1[q] \times F_{2,n-1} \times A_n[q'] \\ i = 1 & 2 < j < n & \Rightarrow m_{1j}(q,q') & = A_1[q] \times F_{2,j-1} \times A_j[q'] \times F_{j+1,n} \\ 1 < i < n-1 & j = n & \Rightarrow m_{in}(q,q') & = F_{1,i-1} \times A_i[q] \times F_{i+1,n-1} \times A_n[q'] \\ 1 < i < n-2 & i+1 < j < n & \Rightarrow m_{ij}(q,q') & = F_{1,i-1} \times A_i[q] \times F_{i+1,j-1} \times A_j[q'] \times F_{j+1,n} \\ i = 1 & j = 2 & \Rightarrow m_{12}(q,q') & = A_1[q] \times A_2[q'] \times F_{3,n} \\ i = n-1 & j = n & \Rightarrow m_{in}(q,q') & = F_{1,n-2} \times A_{n-1}[q] \times A_n[q'] \\ 1 < i < n-1 & i+1 = j & \Rightarrow m_{ij}(q,q') & = F_{1,i-1} \times A_i[q] \times A_{i+1}[q'] \times F_{i+2,n} \end{array}$$

All the $F_{i,j}$ can be computed in an efficient way recursively with the following algorithm:

Algorithm 1 Recursive computation of the F_{ij}

inputs: $B_i \;\; \forall\, 1 \leq i \leq n$
initialisation: $\forall\, 1 \leq i \leq n, \;\; F_{i,i} = B_i$
for $i = 1$ **to** $n-1$ **do**
  for $j = i+1$ **to** n **do**
    $F_{i,j} = F_{i,j-1} \times B_j$
  end for
end for
return $F_{i,j} \;\; \forall\, 1 \leq i < j \leq n$

Once the F_{ij} are computed, all binary marginals can be computed with the following algorithm:

Algorithm 2 Computation of all binary marginals of a pairwise graphical model

inputs: $F_{i,j}\ \forall\, 1 \leq i < j \leq n$, $A_k(q)\ \forall\, 1 \leq k \leq n, \forall\, 1 \leq q \leq Q$
for $j = 1$ **to** $n-1$ **do**
  for $q' = 1$ **to** Q **do**
    $R_j[q'] = A_j[q'] \times F_{j+1,n}$
  end for
end for
for $i = 2$ **to** n **do**
  for $q = 1$ **to** Q **do**
    $L_i[q] = F_{1,i-1} \times A_i[q]$
  end for
end for
for $i = 1$ **to** $n-1$ **do**
  for $j = i+1$ **to** n **do**
    for $q = 1$ **to** Q **do**
      for $q' = 1$ **to** Q **do**
        if $i = 1$ and $j = n$ **then**
          $m_{ij}(q, q') = A_1[q] \times F_{2,n-1} \times A_n[q']$
        else if $i = 1$ and $2 < j < n$ **then**
          $m_{ij}(q, q') = A_1[z_1] \times F_{2,j-1} \times R_j[q']$
        else if $1 < i < n-1$ and $j = n$ **then**
          $m_{ij}(q, q') = L_i[q] \times F_{i+1,n-1} \times A_n[q']$
        else if $1 < i < n-2$ and $i+1 < j < n$ **then**
          $m_{ij}(q, q') = L_i[q] \times F_{i+1,j-1} \times R_j[q']$
        else if $i = n-1$ and $j = n$ **then**
          $m_{ij}(q, q') = L_{n-1}[q] \times A_n[q']$
        else if $1 < i < n-1$ and $i+1 = j$ **then**
          $m_{ij}(q, q') = L_i[q] \times R_{i+1}[q']$
        end if
      end for
    end for
  end for
end for
return $m_{ij}(q, q')\ \forall\, 1 \leq i < j \leq n, \forall\, 1 \leq q, q' \leq Q$

Computing all the $F_{i,j}$ requires $\sum_{i=1}^{n-1}(n-i) = n(n-1)/2$ matrices products. Then, computing all $L_i[q]$ for a given q requires $n-1$ matrices products, and there are Q possible values for q, hence $(n-1)Q$ matrices products are needed to obtain all matrices $L_i[q]$. There is a similar result for all matrices $R_j[z_j]$. Knowing all $F_{i,j}$, $L_i[q]$ and $R_j[q']$, the calculation of $m_{ij}(q, q')$ requires 2 matrices products, hence $n(n-1)Q^2$ products for all binary marginals because there are $n(n-1)/2$ pairs (i, j). In total, computing all binary marginals requires $n(n-1)/2 + 2(n-1)Q + n(n-1)Q^2 \approx n^2Q^2$ matrix multiplications. Let us recall that the real barrier to this calculation is the

dimension of the matrices, hence each matrices product. These dimensions are evaluated in next section.

Note that the procedure is more general and is not restricted to binary marginals. It can also be used to compute marginals of order 3 or more since any marginal can be expressed as products of some B_k and $A_k[z_k]$.

5.3 Dimensions of the matrices $A_k(z_k)$ for a WSBM

If the graphical model is a WSBM, for which $E = \{(i,j) : 1 \leq i < j \leq n\}$, the price to pay in complexity for variable separation using TT is very high. Indeed, the dimensions of the matrices $A_k(z_k)$ are huge. This can be understood intuitively before developing the calculation, as each $A_k(z_k)$ is a Kronecker product of m matrices $G_k^{ij}[z_k]$ with $m = n(n-1)/2$. Each matrix $G_k^{ij}[z_k]$ has dimension 1×1, $1 \times Q$, $Q \times 1$, or $Q \times Q$, and their Kronecker product has dimension at most $Q^m \times Q^m$. Hence, complexity comes from the number of pairs (i,j) in E. What follows is established for WSBM, but the approach is relevant for any pairwise graphical model.

We first establish that, for WSBM models with n individuals, the dimensions of the matrix $A_k(z_k)$ is

$$\dim A_k(z_k) = Q^{(k-1)(n-k+1)} \times Q^{k(n-k)}$$

Proof. Let us start from

$$A_k(z_k) = \bigotimes_{1 \leq i < j \leq n} G_k^{ij}[z_k]$$

(see equation (9) with $(i,j) \equiv \ell$). Let us recall that if $\dim A = (a, a')$ and $\dim B = (b, b')$, then $\dim A \otimes B = (ab, a'b')$. $A_k(z_k)$ is a Kronecker product of $m = n(n-1)/2$ matrices. Let us assume that m_r matrices are in $\mathbb{R}$, m_q in $\mathbb{R}^{1 \times Q}$, $m_{q'}$ in $\mathbb{R}^{Q \times 1}$ and m_Q in $\mathbb{R}^{Q \times Q}$. Then

$$\dim A_k(z_k) = Q^{m_{q'} + m_Q} \times Q^{m_q + m_Q}$$

Let us select a $k \in \{1, \ldots, n\}$ and express $m_q, m_{q'}$ and m_{Q} in terms of k and n. The dimension of matrix $G_k^{ij}[z_k]$ depends on the position of k when compared to i and j according to equation (C14). The value of $G_k^{ij}[z_k]$ depending on i and j can be visualised in figure 1 for $k \notin \{i, j\}$.

k being fixed, the counts of pairs (i,j) per condition in (C14) follow as:

- → condition $k < i$: it corresponds to all pairs (i,j) with $k+1 \leq i < j \leq n$, hence $(n-k)(n-k-1)/2$ pairs
- → condition $k = i$: it corresponds to all pairs (i,j) with $k = i < j \leq n$, hence $n-k$ pairs
- → condition $i < k < j$: it corresponds to all pairs (i,j) with $i < k < j$, hence $k-1$ values for i and $n-k$ for j, hence $(k-1)(n-k)$ pairs
- → condition $k = j$: it corresponds to all pairs (i,j) with $1 \leq i < k = j$, hence $k-1$ pairs
- → condition $k > j$: it corresponds to all pairs (i,j) with $1 \leq i < j < k$, hence $(k-1)(k-2)/2$ pairs

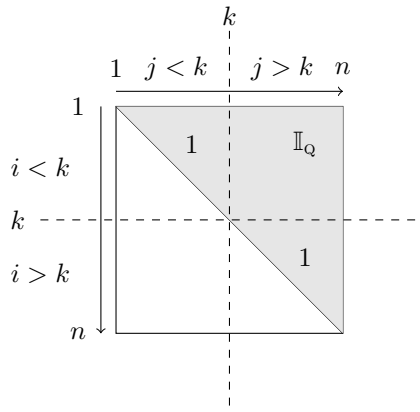


Fig. 1 This matrix displays all matrices $G_k^{ij}[z_k]$ for k being fixed and all pairs $(i,j) \in \{1,\dots,n\} \times \{1,\dots,n\}$. The gray zone represents those for which $1 \leq i < j \leq n$. Indices i for which $k < i$ or $k > i$, and indices j for which $k < j$ or $k > j$ are separated by dashed lines. This enables to show the blocks of indices (i,j) for which $G_{ij}^{(k)} = 1$ and those for which $G_{ij}^{(k)} = \mathbb{I}_Q$

This leads to the following table:

k	number of pairs (i,j)	size of $\overline{\psi_k^{ij}}[z_k]$
$k < i$	$(n-k)(n-k-1)/2$	1×1
$k = i$	$n-k$	$1 \times Q$
$i < k < j$	$(k-1)(n-k)$	$Q \times Q$
$k = j$	$k-1$	$Q \times 1$
$k > j$	$k(k-1)/2$	1×1

Hence, $m_{q'} = k-1$, $m_Q = (k-1)(n-k)$ and $m_q = n-k$. Let us note that the counts for $k < i$ and $k > j$ are not necessary since they correspond to m_r, which is not involved in the size of $A_k(z_k)$. □

The highest values are obtained for $k \simeq n/2$, which yields $\dim A_k(z_k) \simeq Q^{(n/2)^2} \times Q^{(n/2)^2}$, which is worse than Q^n (the number of terms in the tensor of the joint distribution). For example, for $n = 20$ and $Q = 3$, this yields $\dim A_{n/2}[z] \simeq 3^{10^2} = 3^{100} \simeq 5.15 \times 10^{47}$. The growth of the dimension is due to the multiplicity of Kronecker products. Hence, algebra yields exact formula for computing the normalising constant, but leads to inextricable numerical difficulties for the calculation. We address in next section such difficulties. Let us mention that the dimensions are dominated by the pairs (i,j) with $i < k < j$: the highest contribution to the dimension of $A_k(z_k)$ is a Kronecker product of $(k-1)(n-k)$ matrices $\mathbb{I}_Q$. This is the identity matrix of size $Q^{(k-1)(n-k)}$, which is sparse (if $x = Q^{(k-1)(n-k)}$, the number of terms in this identity matrix is x^2, among which, in the diagonal, x are non zero and equal to 1). To the best of our knowledge, taking this sparsity into account to simplify the calculations remains an open question.

6 Numerical difficulties beyond small sized problems

We have seen in section 5.3 that the dimensions of the matrices involved in matrices products for exact computation of the normalising constant or some marginals (like $W = \prod_i B_i$ with $B_i = \sum_{z_i} A_i(z_i)$) have dimensions which can reach $\simeq Q^{(n/2)^2} \times Q^{(n/2)^2}$. It is not possible to perform these calculations using conventional matrix calculation algorithms, in particular the BLAS library. Novikov et al. (2014) proposed to use a TT-format specific for matrix calculation which has been defined in Oseledets et al. (2011) and Oseledets and Dolgov (2012). We call here this format *TT-matrix format*, and not TT-format, because it is not the TT-format of a matrix read as a tensor of order 2 (this TT-format is given by the SVD of the matrix, see section 5.1). To avoid any confusion, we specify "matrix expressed as TT-matrix", or more simply "TT-matrix", instead of matrix in TT format. Calculation can be done exactly in TT-matrix format, because matrices $A_i(z_i)$ are expressed as TT-matrices of rank one, hence matrices B_i are expressed as TT-matrices too, as well as their products. Definition of TT-matrices is given in Section 6.1, after the presentation of multiplexing / demultiplexing of indices. Even when using TT-matrices, numerical difficulties remain and we present in Section 6.2 how we propose to bypass them.

6.1 Matrices expressed as TT-matrices

Multiplexing / demultiplexing of indices:

Let $\mathbf{z} = (z_1, \ldots, z_n) \in \{1, \ldots, Q\}^n$ be a multi-index. An index a in $\{1, \ldots, Q^n\}$ can be associated to $\mathbf{z}$ using the following construction: $a = \sum_{i=1}^{n-1} Q^{n-i}(z_i - 1) + z_n$, which is a bijection between $\{1, \ldots, Q\}^n$ and $\{1, \ldots, Q^n\}$. The transformation from a to $\mathbf{z}$ is called multiplexing, and the reverse operation is called demultiplexing.

Definition of a matrix expressed as a TT-matrix:

Let A be a $Q^n \times Q^n$ matrix. Let $a \equiv (z_1, \ldots, z_i, \ldots, z_n)$ and $b \equiv (t_1, \ldots, t_i, \ldots, t_n)$ be two demultiplexed indices and their multiplexed equivalent. A is said to be expressed in TT-matrix format of rank r if there exists n families of $r \times r$ matrices, where each family is composed of Q^2 matrices and is indexed by i, and where each matrix in a given family is indexed by (z_i, t_i):

$$\forall\, i, \qquad (z_i, t_i) \longrightarrow M_i(z_i, t_j), \qquad \text{with} \quad \begin{cases} i & \in \{1, \ldots, n\} \\ z_i, t_j & \in \{1, \ldots, Q\} \\ M_i(z_i, t_j) & \in \mathbb{R}^{r \times r} \end{cases}$$

such that

$$A[a, b] := A[(z_1, \ldots, z_n)\,;\,(t_1, \ldots, t_n)] = M_1(z_1, t_1) \ldots M_n(z_n, t_n). \tag{11}$$

(M_1 is $1 \times r$ and M_n is $r \times 1$ for the r.h.s. to be a scalar). Matrices $M_i(z_i, t_i)$ are called the "cores" of the TT-matrix. Each matrix $M_i(z_i, t_i)$ has r^2 elements (or r for

$M_1(z_1, z_1)$ and $M_n(z_n, t_n)$), and there are nQ^2 of them. Hence, storage of A in TT-matrix format (i.e. storing the matrices $M_i(z_i, t_i)$ instead of storing A) requires nQ^2r^2 elements instead of Q^n.

Elementary operations of matrix calculus in TT-matrix format:

In addition to the impressive storage savings achieved by using the TT-matrix format when possible, basic matrix calculus operations can be carried out directly in TT-matrix format, without the need for massive and complete storage of matrices in memory. It is possible to compute the cores of the sum $A + B$ of two matrices, or of their product AB (see appendix B), knowing the cores of A and B only. It is possible to show that TT_rank $(A + B) \leq$ TT_rank $A +$ TT_rank B and TT_rank $(AB) \leq$ (TT_rank A)(TT_rank B).

Application to the computation of W: controlling the TT-rank:

So, all calculations of $B_k = \sum_{z_k} A_k(z_k)$, $W = \prod_k B_k$ and marginals can be computed in TT-matrix format. We have TT_rank $B \leq Q$, and TT_rank $W \leq Q^m$. Of course, W is a scalar, and TT_rank $W = 1$. However, the rank of intermediate products can be exponential with the number of terms in the product. For controlling the exponential growth of the TT-rank when computing the product $\prod_i B_i$, algorithm 1 in Novikov et al. (2014) consists in approximating the product by a matrix of low TT-rank, with an operation called the "rounding", which can also be performed in TT-matrix format. Let us recall that each matrix B_i in $W = \prod_{i=1}^{n} B_i$ is the sum of Q matrices $A_i(q)$ for $q \in \{1, \ldots, Q\}$. Hence, storage of a matrix B_i requires a memory space of size mQ^4 ($r = Q$). Using the TT-toolbox, we have carried out the control the TT-rank of the matrices after each product with rounding with a prescribed accuracy of $\epsilon^{TT} = 5.10^{-2}$, starting with $n = 8$ variables and increasing n progressively. At given values of n, some new numerical difficulties appeared. We present them next in the order in which they have appeared, and the choices we made for overcoming each of them.

6.2 Numerical difficulties encountered and options for overcoming them

1. for $n \geq 24$, numerical instabilities occurred because of small values for W or marginals. So we have developed a simple action to fix it (see Calculating with very small values);
2. for $n \geq 30$, the rounding at prescribed accuracy yields matrices with too large TT-rank (the TT-rank is no longer controlled). Therefore, a new strategy has been carried out for the control of the TT-rank (see Strategy for controlling the TT-rank in the rounding);
3. for $n \geq 45$, the number of cores became an issue. Indeed, each matrix B_i has m cores. The rounding requires $\sim m^2$ factorisations LQ and SVD which becomes prohibitive when m is large, and the TT-toolbox is limited to matrices with 1024 cores or less. As the number m of cores is $n(n-1)/2$, this corresponds to $n = 45$ nodes or less. Therefore, we set up core fusion to work with fewer but larger cores (see Fusion of cores).

We therefore have implementation choices that depend on the problem size, as detailed below. This allowed us to calculate the different marginals up to $n = 65$ for a WSBM with an assortative connectivity matrix (see Section 7 for a description of several WSBM connectivity structures).

Calculating with very small values:

Both the un-normalised marginals and the normalising constant, computed separately, are very small (see table 2): numerical simulations have shown that $W \sim \exp^{-m}$. Because of the finite representation of real numbers in IEEE format and the small number of points to represent very small values, calculating marginals leads to errors when $n \geq 24$. This can be solved easily by scaling the WSBM factors with a factor denoted α. Let us denote with an exponent s the scaled factors: $\psi^s_{i,j}(z_i, z_j) = \alpha\, \psi_{i,j}(z_i, z_j)$. We have $\mathbb{P}^s(z, D) = \alpha^m\, \mathbb{P}(z, D)$ because it is a product of m factors. So, $W^s = \sum_z \mathbb{P}^s(z, D) = \alpha^m\, W$. If $m_{i,j}(z_i, z_j)$ is the un-normalised binary marginal of variables (i, j) for states (z_i, z_j), we have $m_{i,j}(z_i, z_j) = \sum_{z \setminus z_i, z_j} \mathbb{P}(z, D)$, so $m^s_{i,j}(z_i, z_j) = \alpha^m\, m_{i,j}(z_i, z_j)$. So, normalised marginals are unchanged by scaling, because $m^s_{i,j}(z_i, z_j)/W^s = \alpha^m\, m_{i,j}(z_i, z_j)/\alpha^m\, W = m_{i,j}(z_i, z_j)/W$. We have chosen $\alpha = 10$, which is not the optimal parameter, but it is sufficient in our case.

Strategy for the controlling the TT-rank in the rounding:

There is indeed a trade off between TT-rank and accuracy: as with Singular Value Decomposition of matrices which provides a best low rank approximation of a matrix, the higher the rank, the better the accuracy. We wish at the same time a low TT-rank for memory issues, and a high accuracy. To choose the best balance between low TT-rank and high precision, we carried out a numerical experiment to compare the quality of the calculation of W with prescribed precision and several choices of prescribed TT-rank. For this experiment, we considered a WSBM with Poisson distributed distances. It means that $\mathbb{P}_\Lambda(D[i, j] \mid Z_i = q, Z_j = q')$ is a Poisson distribution with parameter $\lambda_{qq'}$, and the connectivity matrix Λ is defined by $\Lambda[q, q'] = \lambda_{qq'}$. We used

$$\Lambda = \begin{pmatrix} 2 & 10 & 10 \\ 10 & 3 & 8 \\ 10 & 10 & 4 \end{pmatrix}$$

The result is given in table 2 for different values of n (from $n = 8$ to $n = 32$). To this end, in the TT rounding algorithm we have combined the two approaches of prescribed accuracy and prescribed rank, by setting the accuracy to $\epsilon^{TT} = 10^{-2}$, but limiting the rank growth to $r^{TT}_{max} = 27$. If the final rank reaches 27, that means we have lost some of our prescribed accuracy.

Merging of cores:

To have fewer but larger cores in TT-matrix format, with $m \leq 1024$, the cores of the matrices $A_k(z_k)$ can be merged together thanks to the associativity of the Kronecker product, as seen on this very simple example: $M_1 \otimes M_2 \otimes M_3 \otimes M_4 = (M_1 \otimes M_2) \otimes (M_3 \otimes M_4)$: we have two larger cores as far as dimensions are concerned instead of four

n	R_1	R_2	R_3	R_4	R_5
8	6.21×10^{-28} (9)	2.68×10^{-28}	6.21×10^{-28}	6.21×10^{-28}	6.21×10^{-28}
10	3.2×10^{-45}(12)	2.24×10^{-46}	3.20×10^{-45}	3.2×10^{-45}	3.2×10^{-45}
12	1.0×10^{-75}(31)	3.32×10^{-77}	6.73×10^{-76}	1.0×10^{-75}	1.0×10^{-75}
14	5.38×10^{-101}(8)	1.11×10^{-101}	5.38×10^{-101}	5.38×10^{-101}	5.38×10^{-101}
16	3.30×10^{-135}(10)	3.05×10^{-135}	3.30×10^{-135}	3.30×10^{-135}	3.30×10^{-135}
18	2.05×10^{-161}(15)	1.14×10^{-161}	2.05×10^{-161}	2.05×10^{-161}	2.05×10^{-161}
20	1.18×10^{-199}(18)	2.65×10^{-205}	1.18×10^{-199}	1.18×10^{-199}	1.18×10^{-199}
22	1.59×10^{-247}(15)	1.80×10^{-247}	1.59×10^{-247}	1.59×10^{-247}	1.59×10^{-247}
24	9.57×10^{-294}(16)	1.12×10^{-297}	9.57×10^{-294}	9.57×10^{-294}	9.57×10^{-294}
26	1.80×10^{-353} (16)	8.13×10^{-375}	1.80×10^{-353}	1.80×10^{-353}	1.80×10^{-353}
28	1.34×10^{-393} (21)	1.09×10^{-393}	1.34×10^{-393}	1.34×10^{-393}	1.34×10^{-393}
30	4.03×10^{-459} (21)	4.03×10^{-459}	4.03×10^{-459}	4.03×10^{-459}	4.03×10^{-459}
32	3.43×10^{-511} (34)	3.19×10^{-511}	3.43×10^{-511}	3.43×10^{-511}	3.43×10^{-511}

Table 2 Normalising constant calculated according to different accuracy and maximal rank settings in rounding: R1: $\epsilon^{TT} = 5.10^{-2}$; R2: $r_{\max}^{TT} = 3$; R3: $r_{\max}^{TT} = 9$; R4: $r_{\max}^{TT} = 27$; R5: $r_{\max}^{TT} = 81$. For $\epsilon^{TT} = 5.10^{-2}$, the maximum of observed rank is given in brackets. Calculations have been done for a WSBM with Poisson distributed distances and $Q = 3$.

smaller cores. This has been implemented for each matrix $A_k(z_k)$. Let us recall that (see formula (C15)): $A_k(z_k) = \bigotimes_{1 \leq i < j \leq n} G_k^{ij}(z_k)$ The fusion here is carried out at the level of the matrices $G_k^{ij}[z_k]$. Merging can be implemented either by building groups with the same number of initial cores (uniform numbers), or by creating merged cores of the same dimensions (uniform volumes). We have tested the impact of each method on the calculation of W for different values of n for the same WSBM model than above. (see table 3). As the results are very similar, and observing that the method with uniform numbers is much simpler to implement, we have chosen to merge cores with uniform numbers.

n	Without merging	Merging with uniform numbers	Merging with uniform volumes
12	9.35×10^{-71}	9.35×10^{-71}	9.35×10^{-71}
15	8.04×10^{-112}	7.95×10^{-112}	7.95×10^{-112}
18	3.77×10^{-163}	3.77×10^{-163}	3.77×10^{-163}
21	2.92×10^{-219}	2.92×10^{-219}	2.92×10^{-219}
24	1.53×10^{-281}	1.53×10^{-281}	1.53×10^{-281}
27	3.49×10^{-359}	3.76×10^{-359}	3.73×10^{-359}
30	4.40×10^{-459}	4.40×10^{-459}	4.40×10^{-459}
33	7.81×10^{-565}	7.81×10^{-565}	7.81×10^{-565}
36	1.55×10^{-673}	1.54×10^{-673}	1.55×10^{-673}
39	1.08×10^{-775}	1.08×10^{-775}	1.08×10^{-775}
42	1.06×10^{-915}	1.06×10^{-915}	1.06×10^{-915}

Table 3 Normalising constant without merging cores, or when merging them with uniform numbers or uniform volumes, for increasing number of variables (n). Calculations have been done for a WSBM with Poisson distributed distances and $Q = 3$.

Remark:

The solutions proposed above to overcome the numerical obstacles in carrying out the computation of marginal for large n with separation of variables in TT format have led to the possibility of computing them on data sets with $n \leq 65$. Even though the algebraic calculation is exact, the rounding steps introduce approximations in the numerical implementation.

7 Comparison of the TT-based method and state-of-the-art approaches on simulated data

In this section we compare the behaviour of the TT-based method (refered to as TT in the following) for computing binary marginals in a WSBM model, to that of three classical inference methods: the exact method by complete enumeration, the Gibbs Sampler and the Mean-Field approximation. For these experiments, we considered a WSBM with Poisson distributed distances.

7.1 Different methods for computing the marginals

The first method tested for computing the binary marginals (3) is the complete enumeration of all the terms in the sum. It means Q^{n-2} terms for computing a single binary marginal. So this method is only available for small n. Beyond $n = 12$ it is not possible to store the tensor corresponding to $\mathbb{P}_\theta(Z, D)$ in memory. To circumvent this complexity two strategies are classically used. The exact computation can approximated using simulations of the model, it leads for instance to the Gibbs Sampler (Robert and Casella, 2004). The other option consists in computing the marginals on a simpler model close enough to the original one in a certain sense. This is the principle of the Mean-Field approximation used in the E step of the EM algorithm for WSBM in Daudin et al. (2008). The Gibbs Sampler can be very precise but to the price of a large computing time due to the number of simulations required. On contrary, the Mean-Field method is faster to solve, but the quality of the approximated marginals is lower. Let us detail these two methods.

Gibbs Sampler.

The Gibbs Sampler (GS, Geman and Geman, 1984) is a Markov Chain Monte-Carlo method that exploits conditional probabilities associated to the complex propability of interest. In our case, GS consists in simulating iteratively a Markov chain, using only $\mathbb{P}_\theta(Z_i = q|\{Z_j = z_j\}_{j\neq i}, D)$, and whose stationary distribution converges towards $\mathbb{P}_\theta(Z \mid D)$. In practice, we simulate a single GS run (or path) for a number N^{GS}_{burnin} iterations, then we sample a simulated value of Z every $N^{GS}_{thining}$ new iterations until we collect L realisations. Binary marginals of $\mathbb{P}_\theta(Z|D)$ can be approximated by empirical frequencies.

Mean-Field.

With the Mean-Field approximation (MF), the conditional distribution $\mathbb{P}_\theta(Z \mid D)$ is approximated by a distribution $\mathbb{Q}_\theta(Z \mid D)$ for which marginalisation is less costly.

$\mathbb{Q}_\theta(Z \mid D)$ is chosen among the family $\mathcal{Q}$ of distributions that satisfy the hypothesis of mutual independence between the Z_is. It means that $\mathbb{Q}_\theta(Z \mid D) = \prod_{i=1}^{n} q_\theta^i(Z_i \mid D)$, where the $q_\theta^i(Z_i \mid D)$ are the unary marginals of $\mathbb{Q}_\theta$. The binary marginals are easily obtained as $\mathbb{Q}_\theta(Z_i = q, Z_j = q' \mid D) = q_\theta^i(Z_i = q \mid D) q_\theta^j(Z_j = q' \mid D)$.

The distribution $\mathbb{Q}_\theta(Z \mid D)$ is chosen as the one that minimises the Kullback-Leibler divergence with the true distribution $\mathbb{P}_\theta(Z \mid D)$:

$$\mathbb{Q}_\theta(. \mid D) = \arg\max_{\mathbb{Q} \in \mathcal{Q}} KL(\mathbb{Q}(.) \mid \mathbb{P}_\theta(. \mid D))$$

with $KL(\mathbb{Q}(.)|\mathbb{P}(.)) = \sum_z \mathbb{Q}(z) \ln\left(\frac{\mathbb{Q}(z)}{\mathbb{P}(z)}\right)$. The solution is on the form of a fixed point equation which is solved by an iterative scheme (see Appendix A for the demonstration). If we denote by $\tau_{i,q}^t$ the current value of $\mathbb{Q}(Z_i = q \mid D)$, we have

$$\forall 1 \leq i \leq n, \forall 1 \leq q \leq Q, \quad \begin{cases} \tau_{i,q}^{t+1} = \frac{\mu_{i,q}^{t+1/2}}{\sum_{l=1}^{Q} \mu_{i,l}^{t+1/2}} \\ \mu_{i,q}^{t+1/2} = \alpha_q \prod_{j \neq i} \prod_{q'=1}^{Q} \left[\frac{\lambda_{q,q'}^{D(i,j)}}{D(i,j)!} \exp -\lambda_{q,q'} \right]^{\tau_{j,q'}^t} \end{cases}$$

The scheme is initialised by $\tau_{i,q}^0 = \alpha_q$. We stop the scheme when $\|\Delta\|_\infty < \epsilon^{MF}$ with $\Delta = \{\Delta_{i,q}, 1 \leq i \leq n, 1 \leq q \leq Q\}$ and

$$\Delta_{i,q} = \frac{|\tau_{i,q}^{t+1} - \tau_{i,q}^t|}{|\tau_{i,q}^t|}$$

.

7.2 The 6 WSBM structures

In the literature, different structures of the connectivity matrix Λ has been considered (Kalmbach et al., 2017; Funke and Becker, 2019). They correspond to different organisations of the distances between the individuals. We have selected the following ones: assortative (two examples, easy and difficult), dissortative, core periphery, ordered, and hierarchical structures. The assortative structure corresponds to the situations where the individuals are organised into well separated communities : intra group dissimilarities λ_{qq} are small and inter group dissimilarities $\lambda_{qq'}$ are large. In the dissortative structure, it is the opposite. In a core periphery structure, groups are organised with a core plus groups at increasing distance from this core. Individuals in the core are close to each other while those in the periphery are more and more scattered. In an ordered structure, all the intra group dissimilarities are small then groups have a chain-like organisation where a group is close only to its direct neighbours. In the hierarchical structure, all inter dissimilarities are large and the intra dissimilarities range from small to large. For each structure, the Λ matrix that we used in the experiments, as well as a realisation of the dissimilarity matrix D for $n = 100$ are presented in Figure 2.

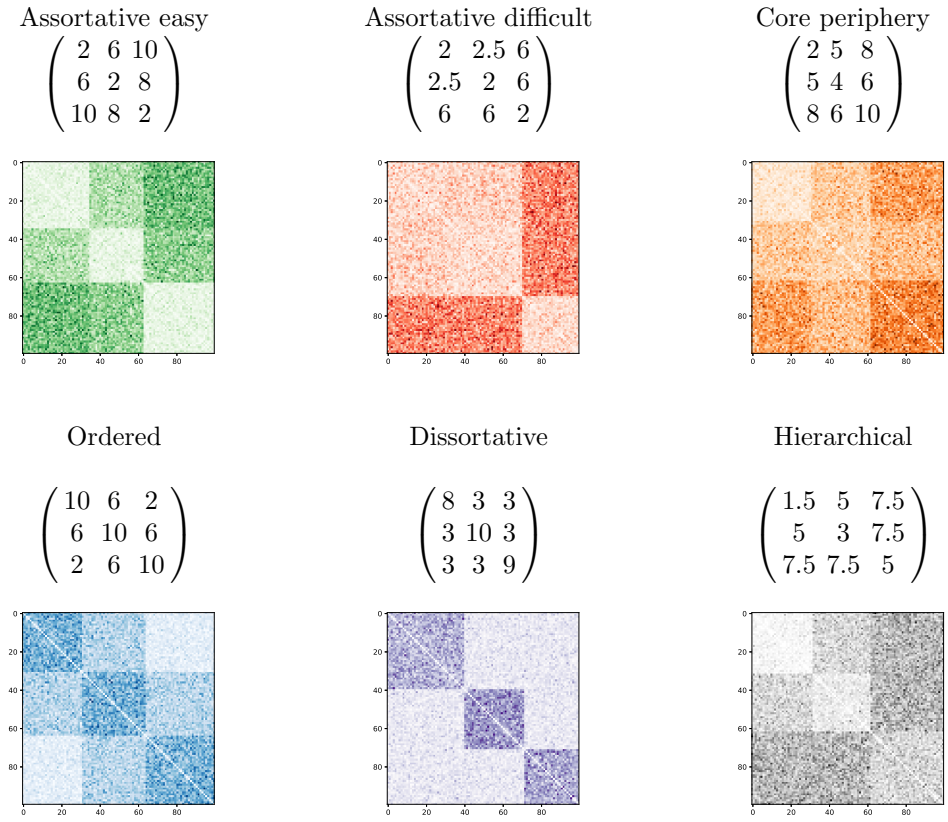


Fig. 2 The 6 WSBM structures used for the numerical experiments: Λ connectivity matrix and a realisation of the dissimilarity matrix D for $n = 100$ individuals, for each structure.

7.3 Protocol

For $n = 12$, exact computation is still possible and so we are able to compare the binary marginals computed by TT, GS and MF to the true ones. For each of the 6 connectivity matrix of Figure 2 we generated 10 dissimilarity matrices D and we ran inference of each of them with the 4 methods. We also ran the comparison for 10 random Λ matrices. Then, for $n = 25$ and $n = 40$, we compared TT, MF and GS behaviours, only on the two assortative structures and the core-periphery structure. We chose the two assortative structures because, they represent typical easy and difficult inference problems. Indeed, even though they belong to the same family of assortative structures, one can see from Figure 2 than the dissimilarity matrix generated using the assortative difficult structure provides much less information on the underlying groups than for the assortative easy structure. This is due to the fact that in the connectivity matrix Λ of the assortative difficult structure, two lines are almost identical, meaning that two groups share almost the same pattern of connections. We also considered the core-periphery structure because for $n = 12$ we observed that the associated unary marginals have a different pattern than for the two assortative

structures (see Section 7.4.1 and Table 4). Therefore we can expect that the 3 selected structures correspond to different difficulties regarding marginal inference.

Generation of the dissimilarity matrices. We set $Q = 3$ and $\alpha_q = 1/3$ in all the experiments. Then, for a given number of individuals n and a given connectivity matrix Λ, we generated a realisation z of $\mathbb{P}_\alpha(Z)$ and then we generated a dissimilarity matrix D according to $\mathbb{P}_\Lambda(D \mid Z = z)$.

Settings of the inference methods. The parameters for each method are set to the following values:

- *exact method*: with this method, the tensor of the joint distribution is stored in memory, and the binary and unary marginals are computed by using expression (3). It does not require any parameter setting;
- *TT*: parameters are the accuracy ϵ^{TT} and the maximal TT-rank r^{TT}_{max}. We set $\epsilon^{TT} = 10^{-2}$ and $r^{TT}_{max} = 27$ for all experiments except $n = 40$ and the assortative easy structure where we lowered r^{TT}_{max} to 9 since it was enough to reach a good agreement with the other methods.
- *Gibbs Sampler*: parameters are the number of iterations $L^{GS} = 5 \times 10^5$, the warming phase $N^{GS}_{burnin} = 10^3$ and the thinning frequency $N^{GS}_{thinning} = 5$. There exist no universal operational rules to set these parameters (Casella and George, 1992). We choose these values empirically by checking that for $n = 12$ the GS estimates of unary marginals converge to the true values in most cases, and that for $n = 25$ or 40 GS reached an agreement with the two other approximate methods;
- *Mean-Field*: accuracy for stopping the iteration: $\epsilon^{MF} = 10^{-1}$.

Note that these choices may be specific to $Q = 3$.

Output. Computation of binary marginals by Gibbs Sampler can be subject to label switching (Murphy, 2012, chapter 24). It means that there should be a phase of relabeling before comparison with the binary marginals obtained with the other methods. To avoid this step which can be tricky, we compared the values of the probability that two individuals i and j belong to the same class. This probability is obtained as $PSC_{i,j} = \sum_{q=1}^{Q} \mathbb{P}_\theta(Z_i = q, Z_j = q \mid D)$. This quantity is not sensible to label switching. Furthermore, in most applications, this quantity is more relevant than the knowledge of each probabilities $\mathbb{P}_\theta(Z_i = q, Z_j = q' \mid D)$.

7.4 Results

7.4.1 Numerical experiments for $n = 12$

In a preliminary step, we compared the value of the exact and approximated unary marginals. The list of all marginals for a dissimilarity matrix is a list 12×3 values, so we were able to compared the exact and approximated values one by one instead or summarising the difference by some statistics. We observed that the shape of the unary marginals varies a lot between the structures, and it can also vary among the 10 repetitions for a given Λ matrix due to the variance of the Poisson distribution. For assortative easy the true unary marginals are all very close to a Dirac, i.e. with

one group of high probability (> 0.9) and the two others of very low probability. For assortative difficult and ordered structures, some marginals are close to Dirac and other are close to a uniform distribution between 2 groups (the third one having a probability of zero). Finally, for core-periphery, hierachic and random, the true unary marginals are more variables. The first observation is that the unary marginals computed using TT always follow the same pattern than the true ones, which is not the case for GS and MF (see Table 4). In addition quite often MF did not converge. Beyond this qualitative observation, quantitatively the TT unary marginals are always almost identical to the true one (up to the 3rd decimal). GS can be good to very good, however the quality varies with the structure, and sometimes between different runs of a same structure. When MF reaches convergence, the unary are not always of same quality than TT or GS.

Regarding now binary marginals, Table 5 provides, for each Λ structure the median of the absolute difference between the value of $PSC_{i,j}$ computed with the exact method and each of the approximate ones. We chose the median rather than the mean since histograms of these absolute difference are often bimodal (see Figure 1 in the SI). The value reported is the mean of this median over the 10 repetitions. Similarly Table 6 displays the mean computational time, over the 10 repetitions, for computing all unary and binary marginals, for each Λ structure.

Having in mind that the true value of $PSC_{i,j}$ is between 0 and 1, the global conclusion from Table 5 is that the three approximate methods provide good to very good estimates, and TT almost always leads to the lowest error. The MF estimator is also precise but less than TT. Furthermore, as above-mentioned, it can happen that the iterative scheme for solving the fixed point equation does not converge. This concerned between 0 and 10 repetitions depending on the Λ structure (see Table 5). We also observed that the MF approximation of $PSC_{i,j}$ is of very good quality when the unary marginals are close to 0 or 1. This occurs typically with the 'assortative easy' structure. In this case, the MF approximates very well the unary marginals. Since it can be shown (see Appendix C) that when unary marginals are equal to 0 or 1 then the binary marginals are equal to the product of the unary ones, mechanically MF leads to a good approximation of $PSC_{i,j}$ in this case. If the unary marginals are not close to 0 or 1, the quality if the MF approximation can decrease, this is the case for instance for the random structure.

As for the unary marginals, we observed that the quality of the approximation can vary with the structure of Λ. For instance, whatever the method, the approximation is of better quality for the "assortative easy" structure than the "assortative difficult" structure. For the latter, group 1 and group 2 have very similar connectivity pattern, therefore they can be difficult to distinguish from the dissimilarity matrix D.

Note that a method can provide a good approximation of PSC while leading to a poor approximation of the unary marginals. This is due to the fact that the computation of $PSC(i,j)$ does not use all the binary marginals $\mathbb{P}_\theta(Z_i = q, Z_j = q' \mid D)$ but only those where $q = q'$. On contrary, all binary marginals are required to compute the unary marginals. It happends for instance for GS with the ordered structure or MF with the dissortative structure, so it means that in this cases GS and MF did not approximate well the binary marginals $\mathbb{P}_\theta(Z_i = q, Z_j = q' \mid D)$ when $q \neq q'$.

Regarding computational time, the four methods rank as expected, with from the faster to the slowest MF, TT, GS. For TT and MF, the computational time varies with the structure of Λ.

7.4.2 Numerical experiments for $n > 12$

For $n = 25$ and $n = 40$, we compared the different methods on three structures which present three different patterns of unary marginals on the experiments for $n = 12$: the two assortative structures and the core-periphery one. Table 7 provides the median of the absolute difference between the value of $PSC_{i,j}$ computed with two pairs of approximate methods. The median is computed over all pairs of variables i, j, and the value reported in Table 7 is the mean of this median value, over the 10 experiments. Table 8 displays the mean computational time (over the 10 experiments) for computing all unary and binary marginals.

The agreement between the TT, GS and MF estimates is very good for the assortative easy structure. In this case, the unary marginals are close to 0/1, so, as explained previously, it is a situation where MF performs well. The agreement between MF and the two other methods increases when $n = 40$, probably because, heuristically, Mean-Field approximation originating from statistical physics is more appropriate for systems with a large number of interacting sites.

Agreement between the three methods is weaker for the assortative difficult structure than for the assortative easy one. Still, it remains of good quality. At first sight, TT and MF are closer to each other than GS is with them. However, when looking at the biplots of all $PSC_{i,j}$ for all pairs (i, j) we observe that MF quite systematically returns values equal to 0, 1 or 1/2, while TT and GS provide more variable values (see Figure 3).

For the core-periphery structure, the agreement between the three method on the value of PSC, while lower than for thee assortative easy structure, is still very good.

Computing time obviously increases with n. MF remains very fast (less than a second). However, as for $n = 12$, the MF does not always converge (see Table 8). So even though it is the fastest method, it has to be used with care. TT remains faster than GS, however the running time of TT can vary a lot with the nature of the connectivity matrix Λ, as opposed to GS and MF. Indeed for the assortative difficult structure we had to change r_{max}^{TT} to 27 for $n = 40$ (instead of 9 for assortative easy). This was necessary to reach a good agreement with GS and MF, however rounding is the demanding task in the TT algorithm and increasing r_{max}^{TT} has a high impact on computing time. Note that we were able to run TT of the assortative easy structure for $n = 50$, but not for the assortative difficult one. The problem was probably due to a saturation of memory in the underlying fortran layer.

8 Conclusions and perspectives

In this paper, we propose an algebraic approach to compute exactly the marginals (normalising constant, unaries, binaries, ternaries, ...) of a pairwise graphical model (the size of the factors is at most two). For this purpose, we have adapted a previous calculation of Novikov et al. (2014) developed for approximating normalising constant

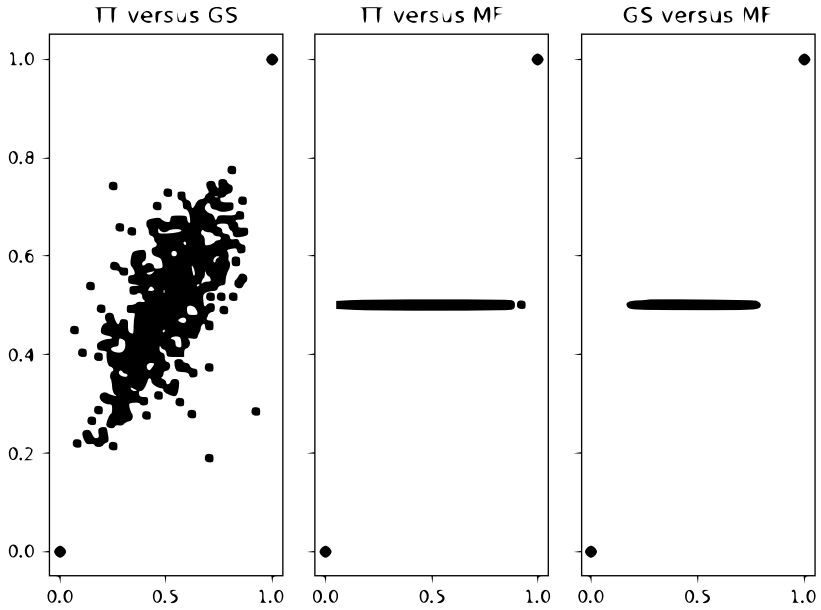


Fig. 3 Biplots of the approximated values of the probability to be in the same group $PSC_{i,j}$ per pair of methods ($Q = 3$, $n = 40$). A dot corresponds to a pair of variables (i, j) with $i < j$. Example for the assortative difficult structure of the WSBM connectivity matrix.

and unary marginals of any graphical model through a clever combination of TT low rank approximations of the factors. Our algebraic calculation leads to the separation of variables in the joint distribution which, by simple distributivity, allows the exact calculation of the marginals through some products of matrices. However, the size of matrices involved grows exponentially with the number m of factors in the graphical model: we have exchanged a number of sums exponential in n with products of sparse matrices of sizes exponential in m. Calculations of such products cannot be done exactly numerically. We then considered numerical approximations in order to practically implement them in the case of the Weighted Stochastic Block Model (WSBM) where the underlying graph is a clique, therefore corresponding to the most complex situation. Such an implementation requires the use of specific techniques, the main ones being working with TT-matrices, and rounding for controlling the TT-rank in matrix $\times$ matrices products.

One of the key issue for controlling the complexity of the numerical approximations is controlling the TT-rank of product of TT-matrices with rounding. What are its quality and efficiency ? It appeared in our numerical simulations that for reasonable sizes of n, like $n = 12$, the rounding works well, i.e. the approximate marginals are of good quality when compared to exact values, and computation times are short, whatever the structure of the connectivity matrix Λ of the WSBM. Further numerical simulations for $n = 25$ and 40 on three structures (called assortative easy, assortative difficult and core-periphery) showed that two elements have an impact on the computation time of the TT approach : the number n of the nodes and the easiness or

difficulty with which some groups can be distinguished. The structure of Λ has also an impact on the precision (from our experiments we cannot determine if n as well because the true marginals are only available for $n = 12$). In structure assortative difficult, two groups are hard to distinguish since they have almost the same connection pattern, whereas in structure assortative easy, all groups are easily distinguished (see Figure 2). TT performs better on assortative easy than on assortative difficult, in the sense that it obtains results much closer to the GS and MF results. Then, in the current version of the implementation, TT cannot be ran on problems with $n = 50$, due to numerical limits in the rounding procedure.

We have compared our TT-based approach with classical approximation approaches like Gibbs Sampler (GS) or Mean-Field approximation (MF). GS has a sound statistical background which guarantees convergence to an unbiased estimator of the marginals (Geman and Geman, 1984; Häggström, 2002), but there are to our knowledge nor theoretical results to tune the parameters for securing a prescribed accuracy, neither generally adopted ad-hoc technique to decide for the number of iterations. Furthermore it may lead to long calculation times, and we observe in practice that it is longer than TT. MF is often quick and accurate for computing unary marginals (Daudin et al., 2008), but it is not designed to compute binary marginals (apart from an coarse estimate as a product of independent unary marginals). It relies on a fixed point solution of some nonlinear system, which sometimes does not converge. TT has an intermediate running time and showed good accuracy on our experiments. So the alternative solution we propose here, based on TT approximation, appears as a useful solution for computing WSBM marginals for intermediate values of n (below 50).

The algebraic exactness of the TT approach (separation of variables) is a very good basis for developing efficient and accurate numerical approaches for calculating the marginals of pairwise graphical models. One challenge for further studies is to go beyond the current numerical limits, moving towards higher sizes, and addressing structures where certain groups are more difficult to distinguish. This should be achieved through investing efforts for repelling the numerical limits in the "rounding" step, possibly by incorporating recent developments in this field, be it with implementations with distributed memory (Al Daas et al., 2022) or, even more recently, with randomised algorithms (Al Daas et al., 2023).

Acknowledgment

Some experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see https://www.plafrim.fr), and on cluster "Hawai" of BioGeCo Research Unit at Pierroton, supported by European Grid Infrastructure programme EOSC-Pillar. M.A. Abouabdallah PhD was funded by INRAE and by INRIA.

9 Author contributions

A.F., N.P., O.C. and M.A.A developed the methodology. M.A.A developed the code. A.F., N.P., O.C. and M.A.A designed the numerical experiments. M.A.A. and A.F. ran the numerical experiments. A.F. and N.P. wrote the manuscript. All authors commented on and approved the final manuscript.

Disclosure statement

The authors report that there are no competing interests to declare.

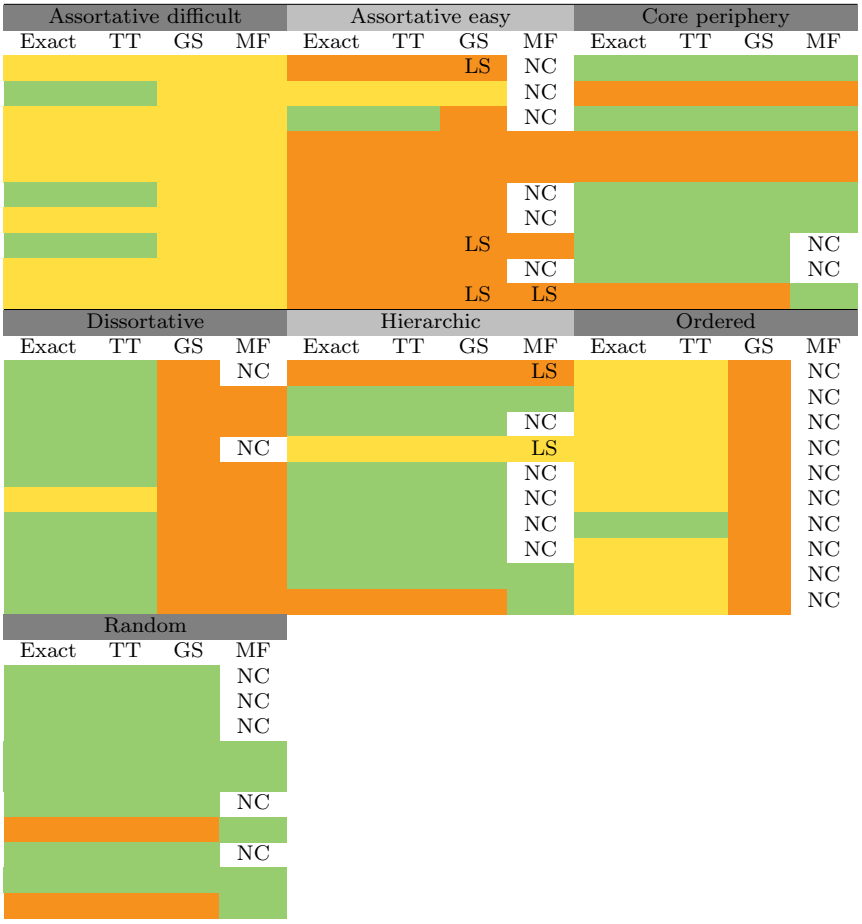


Table 4 Pattern of unary marginals for different structures of the connectivity matrix ($n = 12$ and $Q = 3$). For each structure, 10 dissimilarity matrices have been generated (corresponding to the 10 lines below each structure name) and for each of them the unary marginals of each variable have been computed by 4 different methods (corresponding to the columns in each structure). Then for a dissimilarity matrix and a method we have classified the pattern of the n unary marginals : orange cells correspond to situations where all marginals are Dirac, yellow cells to situations where marginals are either Dirac or uniform on two groups, and green cells to situations where marginals are variable. We also indicate when Label Switching occured (LS) and when MF did not converge (NC). For instance, for the first dissimilarity matrix of structure dissortative, unary marginals computed exactly or with TT were variables (e.g. $(0.6, 0.3, 0.1)$), while the one computed with GS were all Dirac and MF did not converge.

Λ	TT	Gibbs	MF
Assortative difficult	2.126641e-03	1.137171e-01	2.096444e-02 (10)
Assortative easy	7.487330e-10	8.200865e-09	1.177558e-08 (4)
Core periphery	1.430457e-03	1.981476e-04	2.106457e-02 (8)
Dissortative	2.037286e-07	2.723249e-07	7.601169e-07 (8)
Hierarchic (II)	3.293547e-03	6.799929e-03	2.340016e-02(5)
Ordered	3.891483e-08	2.265697e-06	- (0)
Random	2.067789e-02	1.037811e-03	1.925229e-01 (5)

Table 5 Absolute difference between the probability to be in the same group computed by the exact method and by three different approximate methods, for $n = 12$ and $Q = 3$. The value reported is the mean (over 10 runs) of the median of the absolute errors for the $n(n-1)/2$ pairs of variables. For MF, we provide under parentheses the number of runs that converged.

Λ	Exact	TT	Gibbs	MF
Assortative difficult	0.1748605	7.366531	354.3655	0.009956175 (10)
Assortative easy	0.1761076	3.626953	352.9239	0.046608290 (4)
Core periphery	0.1770441	5.178768	368.2358	0.062990053 (8)
Dissortative	0.1755904	4.698245	357.3765	0.059073172 (8)
Hierarchic (II)	0.1756495	7.124235	364.6317	0.021913800 (5)
Ordered	0.1800913	3.962546	359.0718	- (0)
Random	0.1784855	6.781812	361.7040	0.020079037 (5)

Table 6 Computing time (in seconds) for computing all unary and binary marginals, for $n = 12$ and $Q = 3$. The value reported is the mean (over 10 runs). For MF, we provide under parentheses the number of runs that converged.

Λ	$n = 25$			$n = 40$		
	TT vs GS	TT vs MF	GS vs MF	TT vs GS	TT vs MF	GS vs TT
Assort. easy	0	3.8e-16	4.1e-16	9.07e-61	5.03e-49	6.16e-56
Assort. difficult	1.02e-01	2.16e-02	1.53e-01	1.15e-01	1.93e-02	1.86e-01
Core periphery	1.26e-06	4.91e-07	1.54e-06	3.76e-09	1.88e-11	5.73e-10

Table 7 Absolute difference between the probability to be in the same group computed by two approximate methods, for $n = 25$ and $n = 40$, and $Q = 3$. The value reported is the mean (over 10 runs) of the median of the absolute errors for the $n(n-1)/2$ pairs of variables. For TT, experiments on the assortative easy structure were run with $r_{max}^{TT} = 9$ and those on the assortative difficult and core-periphery structures were run with $r_{max}^{TT} = 27$.

	$n = 25$			$n = 40$		
Λ	TT	GS	MF	TT	GS	MF
Assort. easy	61.90	1536.32	0.15 (5)	202.97	4670.47	0.42 (2)
Assort. difficult	278.48	1503.69	0.03 (10)	2790.38	4670.25	0.09 (10)
Core-periphery	86.39	1505.00	0.13 (6)	393.33	4526.13	0.36 (7)

Table 8 Computing time (in seconds) for computing all unary and binary marginals, for $n = 25$ and $n = 40$, and $Q = 3$. The value reported is the mean (over 10 runs). For MF, we provide under parentheses the number of runs that converged. For TT, experiments on the assortative easy structure were run with $r_{max}^{TT} = 9$ and those on the assortative difficult and core -periphery structures were run with $r_{max}^{TT} = 27$.

Appendix A Mean-Field fixed point equation

A.1 Principle

The Mean-Field approximation consists in approximating $\mathbb{P}_\theta(Z \mid D)$ by another joint distribution $\mathbb{Q}_\theta(Z \mid D)$ for which computation of marginals is easier. Under the distribution $\mathbb{Q}_\theta$ the variables Z_i are independent, i.e. $\mathbb{Q}_\theta(Z \mid D) = \prod_{i=1}^n q_\theta^i(Z_i \mid D)$ where $q_\theta^i(Z_i \mid D)$ is the unary marginal of Z_i under $\mathbb{Q}_\theta$. The distribution $\mathbb{Q}_\theta$ is the one that miminises the Kullback-Leibler divergence between a distribution of independent variables and $\mathbb{P}_\theta(Z \mid D)$. We recall the definition of the Kullback-Leibler divergence between two distributions $\mathbb{Q}$ and $\mathbb{P}$:

$$KL(\mathbb{Q}|\mathbb{P}) = \mathbb{E}_\mathbb{Q}\left[\ln\left(\frac{\mathbb{Q}(Z)}{\mathbb{P}(Z)}\right)\right]$$

A.2 Mean-Field as solution of a fixed point equation

Let us define the family $\mathcal{Q}$ of distributions $\mathbb{Q}$ such that $\mathbb{Q}(Z) = \prod_{i=1}^n q^i(Z_i)$, and let us denote $\tau_{i,q} = q^i(Z_i = q)$. Then $\mathbb{Q}_\theta(. \mid D)$ is solution of

$$\mathbb{Q}_{\theta(.|D)} = \arg\min_{\mathbb{Q}\in\mathcal{Q}} KL(\mathbb{Q} \mid \mathbb{P}_\theta(. \mid D))$$

We have

$$KL(\mathbb{Q}(.) \mid \mathbb{P}_\theta(. \mid D)) = \mathbb{E}_\mathbb{Q}\left[\ln(\mathbb{Q})\right] - \mathbb{E}_\mathbb{Q}[\ln(\mathbb{P}_\theta(Z \mid D))].$$

From equations (1) and (2), it is equal to

$$\begin{aligned}
KL(\mathbb{Q}(.) \mid \mathbb{P}_\theta(. \mid D)) &= \sum_{i=1}^n\sum_{q=1}^Q \tau_{i,q}\ln(\tau_{i,q}) - \mathbb{E}_\mathbb{Q}\left[\ln\left(\frac{1}{W}\left(\prod_{i=1}^n\prod_{i>j}\mathbb{P}_\Lambda(D[i,j] \mid Z_i, Z_j)\right)\prod_{i=1}^n\mathbb{P}_\alpha(Z_i)\right)\right] \\
&= \sum_{i=1}^n\sum_{q=1}^Q \tau_{i,q}\ln(\tau_{i,q}) - \mathbb{E}_\mathbb{Q}\left[\ln\left(\frac{1}{W}\times\prod_{i=1}^n\left[\prod_{j>i}\mathbb{P}_\Lambda(D[i,j] \mid Z_i, Z_j)\prod_{q=1}^Q\alpha_q^{Z_{i,q}}\right]\right)\right] \\
&= \sum_{i=1}^n\sum_{q=1}^Q \tau_{i,q}\ln(\tau_{i,q}) - \sum_{i=1}^n\sum_{j>i}\sum_{q=1}^Q\sum_{q'=1}^Q \tau_{i,q}\tau_{j,q'}\ln\mathbb{P}_\Lambda(D[i,j] \mid Z_i = q, Z_j = q') \\
&\qquad - \sum_{i=1}^n\sum_{q=1}^Q(\tau_{i,q}\ln(\alpha_q)) - \ln(\frac{1}{W})
\end{aligned}$$

The minimisation of $KL(\mathbb{Q}(.) \mid \mathbb{P}_\theta(. \mid D))$ is under the constraint that for all i, $\sum_{q=1}^Q \tau_{i,q} = 1$. So, using Lagrangian multipliers, we minimise

$$G = KL(\mathbb{Q} \mid \mathbb{P}(. \mid D)) + \sum_{i=1}^n \sigma_i\left(\sum_{q=1}^Q \tau_{i,q} - 1\right)$$

The variables are the $\tau_{l,q}$ for $l \in \{1, \ldots, n\}$ and $q \in \{1, \ldots Q\}$, and the σ_l for $l \in \{1, \ldots, n\}$. Let us compute the derivatives of G.

$$\frac{\partial G}{\partial \sigma_l} = \sum_{q=1}^{Q} \tau_{l,q} - 1$$

So $\frac{G}{\partial \sigma_l} = 0$ leads to

$$\sum_{q=1}^{Q} \tau_{l,q} = 1 \tag{A1}$$

Then using the fact that D and Λ are symmetric matrices, we obtain:

$$\begin{aligned}\frac{\partial G}{\partial \tau_{l,q}} &= \ln(\tau_{l,q}) + 1 + \sigma_l - \ln(\alpha_q) - \sum_{j=1}^{l-1} \sum_{q'=1}^{Q} \tau_{j,q'} \left[\ln \mathbb{P}_\Lambda(D[j,l] \mid Z_l = q, Z_j = q')\right] \\ &\quad - \sum_{j>l} \sum_{q'=1}^{Q} \tau_{j,q'} \left[\ln \mathbb{P}_\Lambda(D[l,j] \mid Z_l = q, Z_j = q')\right] \\ &= \ln(\tau_{l,q}) + 1 + \sigma_l - \ln(\alpha_q) - \sum_{j \neq l} \sum_{q'=1}^{Q} \tau_{j,q'} \left[\ln \mathbb{P}_\Lambda(D[l,j] \mid Z_l = q, Z_j = q')\right]\end{aligned}$$

So $\frac{\partial g(\sigma_l, \tau_{l,q})}{\partial \tau_{l,q}} = 0$ leads to

$$\tau_{l,q} = \alpha_q \exp\left(\sum_{j \neq l} \sum_{q'=1}^{Q} \tau_{j,q'} \left[\ln \mathbb{P}_\Lambda(D[l,j] \mid Z_l = q, Z_j = q')\right]\right) \exp -(\sigma_l + 1) \tag{A2}$$

We define

$$\mu_{l,q} = \alpha_q \prod_{j \neq l}^{n} \prod_{q'=1}^{Q} \left[\mathbb{P}_\Lambda(D[l,j] \mid Z_l = q, Z_j = q')\right]^{\tau_{j,q'}}$$

Then

$$\tau_{l,q} = \mu_{l,q} e^{-(\sigma_l + 1)} \tag{A3}$$

Finally, by combining equation (A1) and equation (A3) we obtain

$$\tau_{l,q} = \frac{\mu_{l,q}}{\sum_{q=1}^{Q} \mu_{l,q}} \tag{A4}$$

So the $\tau_{l,q}$ are solution of a fixed point equation, since $\mu_{l,q}$ is a function of the $\tau_{j,q'}$ for $j \neq l$.

The associated fixed point iteration scheme is

$$\forall 1 \leq l \leq n, \forall 1 \leq q \leq Q, \quad \begin{cases} \tau_{l,q}^{t+1} = \frac{\mu_{l,q}^{t+1/2}}{\sum_{q'=1}^{Q} \mu_{l,q'}^{t+1/2}} \\ \mu_{l,q}^{t+1/2} = \alpha_q \prod_{j \neq l} \prod_{q'=1}^{Q} \left[\mathbb{P}_\Lambda(D[l,j] \mid Z_l = q, Z_j = q')\right]^{\tau_{j,q'}^{t}} \end{cases}$$

In the case of a Poisson distribution of the dissimilarities, we obtain

$$\forall 1 \leq l \leq n, \forall 1 \leq q \leq Q, \quad \begin{cases} \tau_{l,q}^{t+1} = \frac{\mu_{l,q}^{t+1/2}}{\sum_{q'=1}^{Q} \mu_{l,q'}^{t+1/2}} \\ \mu_{l,q}^{t+1/2} = \alpha_q \prod_{j \neq l} \prod_{q'=1}^{Q} \left[\frac{\lambda_{q,q'}^{D(l,j)}}{D(l,j)!} \exp -\lambda_{q,q'} \right]^{\tau_{j,q'}^{t}} \end{cases}$$

Appendix B Matrix product in TT-matrix format

Let A and B be two matrices with dimensions such that the product $C = AB$ exists. Here, we show how to compute the cores of C expressed in TT-matrix format knowing only the cores of A and B expressed in TT-matrix format. Although it is classical, it is seldom presented in articles. We develop the calculation in detail on a simple example with 3 cores, and evaluate its complexity in a more general case.

Let A, B be TT-matrices with

$$A[(i_1, i_2, i_3); (j_1, j_2, j_3)] = M_1(i_1, j_1)\, M_2(i_2, j_2)\, M_3(i_3, j_3) \tag{B5}$$

and

$$B[(i_1, i_2, i_3); (j_1, j_2, j_3)] = N_1(i_1, j_1)\, N_2(i_2, j_2)\, N_3(i_3, j_3) \tag{B6}$$

with $1 \leq i_\mu, j_\mu \leq n$. Then

$$\begin{aligned} C[(i_1, i_2, i_3); (k_1, k_2, k_3)] &= \sum_{j_1, j_2, j_3 = 1}^{n} A[(i_1, i_2, i_3); (j_1, j_2, j_3)]\; B[(j_1, j_2, j_3); (k_1, k_2, k_3)] \\ &= \sum_{j_1, j_2, j_3 = 1}^{n} M_1(i_1, j_1)\, M_2(i_2, j_2)\, M_3(i_3, j_3)\; N_1(j_1, k_1)\, N_2(j_2, k_2)\, N_3(j_3, k_3) \end{aligned} \tag{B7}$$

Would the matrices product be commutative, reordering the terms would lead to simplifications. But it is not the case. However, reordering can be done in $\mathbb{R}$ between coefficients. Therefore, let us introduce the following notations:

Matrix	size	coefficients
$M_1(i_1, j_1)$	$1 \times r$	$M_1[i_1, j_1, \alpha_1]$
$M_2(i_2, j_2)$	$r \times r$	$M_2[\alpha_1, i_2, j_2, \alpha_2]$
$M_3(i_3, j_3)$	$r \times 1$	$M_3[\alpha_2, i_3, j_3]$
$N_1(j_1, k_1)$	$1 \times r$	$N_1[j_1, k_1, \beta_1]$
$N_2(j_2, k_2)$	$r \times r$	$N_2[\beta_1, j_2, k_2, \beta_2]$
$N_3(j_3, k_3)$	$r \times 1$	$N_3[\beta_2, j_3, k_3]$

with $1 \leq \alpha_i, \beta_j \leq r$. Then

$$\begin{aligned} &M_1(i_1, j_1)\, M_2(i_2, j_2)\, M_3(i_3, j_3)\; N_1(j_1, k_1)\, N_2(j_2, k_2)\, N_3(j_3, k_3) \\ &\qquad = \sum_{\alpha_1, \alpha_2 = 1}^{r} \sum_{\beta_1, \beta_2 = 1}^{r} M_1[i_1, j_1, \alpha_1]\, M_2[\alpha_1, i_2, j_2, \alpha_2]\, M_3[\alpha_2, i_3, j_3] \times \end{aligned}$$

$$N_1[j_1,k_1,\beta_1]\,N_2[\beta_1,j_2,k_2,\beta_2]\,N_3[\beta_2,j_3,k_3] \tag{B8}$$

Let us reorder the terms after the $\sum$ by grouping (α_1,β_1) and (α_2,β_2) as

$$M_1[i_1,j_1,\alpha_1]\,N_1[j_1,k_1,\beta_1]\,M_2[\alpha_1,i_2,j_2,\alpha_2]\,N_2[\beta_1,j_2,k_2,\beta_2]\,M_3[\alpha_2,i_3,j_3]N_3[\beta_2,j_3,k_3] \tag{B9}$$

and define, with γ_i coding for (α_i,β_i) with $\gamma_i = r(\alpha_i-1)+\beta_i$,

$$\underbrace{M_1[i_1,j_1,\alpha_1]\,N_1[j_1,k_1,\beta_1]}_{=P_1[i_1,j_1,k_1,\gamma_1]}\ \underbrace{M_2[\alpha_1,i_2,j_2,\alpha_2]\,N_2[\beta_1,j_2,k_2,\beta_2]}_{=P_2[\gamma_1,i_2,j_2,k_2,\gamma_2]}\ \underbrace{M_3[\alpha_2,i_3,j_3]N_3[\beta_2,j_3,k_3]}_{=P_3[\gamma_2,i_3,j_3,k_3]} \tag{B10}$$

Let us note that, as indices α_i,β_i run each over $\{1,\dots,r\}$, indices γ_i run over $\{1,\dots,r^2\}$. Then

$$\begin{aligned}
&C[(i_1,i_2,i_3);(k_1,k_2,k_3)]\\
&\quad=\sum_{j_1,j_2,j_3=1}^{n}\ \sum_{\gamma_1,\gamma_2=1}^{r^2} P_1[i_1,j_1,k_1,\gamma_1]\,P_2[\gamma_1,i_2,j_2,k_2,\gamma_2]\,P_3[\gamma_2,i_3,j_3,k_3]\\
&\quad=\sum_{\gamma_1,\gamma_2=1}^{r^2}\ \sum_{j_1,j_2,j_3=1}^{n} P_1[i_1,j_1,k_1,\gamma_1]\,P_2[\gamma_1,i_2,j_2,k_2,\gamma_2]\,P_3[\gamma_2,i_3,j_3,k_3]\\
&=\sum_{\gamma_1,\gamma_2=1}^{r^2}\left(\sum_{j_1=1}^{n} P_1[i_1,j_1,k_1,\gamma_1]\right)\left(\sum_{j_2=1}^{n} P_2[\gamma_1,i_2,j_2,k_2,\gamma_2]\right)\left(\sum_{j_3=1}^{n} P_3[\gamma_2,i_3,j_3,k_3]\right)
\end{aligned} \tag{B11}$$

Let us define Q_1,Q_2,Q_3 by

$$\begin{cases}
Q_1[i_1,k_1,\gamma_1] &= \displaystyle\sum_{j_1=1}^{n} P_1[i_1,j_1,k_1,\gamma_1]\\
Q_2[\gamma_1,i_2,k_2,\gamma_2] &= \displaystyle\sum_{j_2=1}^{n} P_2[\gamma_1,i_2,j_2,k_2,\gamma_2]\\
Q_3[\gamma_2,i_3,k_3] &= \displaystyle\sum_{j_3=1}^{n} P_3[\gamma_2,i_3,j_3,k_3],
\end{cases} \tag{B12}$$

with

Matrix	Size	Coefficients	as a tensor
$Q_1(i_1,k_1)$	$1\times r^2$	$Q_1(i_1,k_1)[\gamma_1]$	$=Q_1[i_1,k_1,\gamma_1]$
$Q_2(i_2,k_2)$	$r^2\times r^2$	$Q_2(i_2,k_2)[\gamma_1,\gamma_2]$	$=Q_2[\gamma_1,i_2,k_2,\gamma_2]$
$Q_3(i_3,k_3)$	$r^2\times 1$	$Q_3(i_3,k_3)[\gamma_2]$	$=Q_3[\gamma_2,i_3,k_3]$.

Then

$$C[(i_1,i_2,i_3);(k_1,k_2,k_3)] = \sum_{\gamma_1,\gamma_2=1}^{r^2} Q_1[i_1,k_1,\gamma_1]\, Q_2[\gamma_1,i_2,j_2,\gamma_2]\, Q_3[\gamma_2,i_3,k_3] \\ = Q_1(i_1,k_1)\, Q_2(i_2,k_2)\, Q_3(i_3,k_3), \tag{B13}$$

which is a the expression of C as a TT-matrix of rank r^2.

This calculation can be extended to TT-matrices with more than 3 cores. Let us recall that, in general, A, B have m cores, labelled by μ with $1 \leq \mu \leq m$, of size $r \times r$ (except for the first and last one), and that

$$\begin{cases} A[(i_1,\ldots,i_m);(j_1,\ldots,j_m)] = \prod_{\mu=1}^{m} M_\mu(i_\mu,j_\mu) \\ B[(i_1,\ldots,i_m);(j_1,\ldots,j_m)] = \prod_{\mu=1}^{m} N_\mu(i_\mu,j_\mu), \end{cases}$$

with $1 \leq i_\mu, j_\mu \leq n$. Then, there are n^2 matrices $Q_\mu(i_\mu,k_\mu)$ indexed by (i_μ,k_μ) for a given μ. A matrix $Q_\mu(i_\mu,k_\mu)$ has r^2 terms, each being a sum of n terms of a matrix $P_\mu(i_\mu,j_\mu,k_\mu)$ (see equation (B12)), each term of a P_μ being a product of two real coefficients. So, there are n products for a term of a $Q_\mu(i_\mu,k_\mu)$, nr^2 products for a matrix $Q_\mu(i_\mu,j_\mu)$, and n^3r^2 products for all matrices $Q_\mu(i_\mu,k_\mu)$ for a given μ, and mr^2n^3 products for all matrices $Q_\mu(i_\mu,k_\mu)$ for all cores μ. Hence, the complexity of the matrix product in TT-format is in $\mathcal{O}(mr^2n^3)$. This is for the product of 2 matrices. The complexity of the product of p matrices is in $\mathcal{O}(mr^pn^3)$, and grows exponentially with p. Each matrix A, B has dimensions $n^m \times n^m$, hence their product requires in general $(n^m)^3 = n^{3m}$ products, versus mr^2n^3 in TT-matrix format.

Appendix C Computation of matrices $A_i(z_i)$ for a WSBM

The expression of matrices $A_k(z_k)$ is given here specifically for a WSBM, with two steps, as for general graphical model (see section 4): (i) adding non essential variables, and (ii) computation of matrices $A_k(z_k)$ with mixed product property of Kronecker product.

Adding non essential variables.

Let us show it first on a toy example, with $n = 4$, $i = 2$ and $j = 4$ and E being $\{(i,j) \,:\, 1 \leq i < j \leq n\}$. We have

$$\psi_{24}(z_2,z_4) = M_{24}[z_2]\, V_{42}[z_4] \qquad \text{with} \quad \begin{cases} M_{24}[z_2] & \in \mathbb{R}^{1\times Q} \\ V^{24}[z_4] & \in \mathbb{R}^{Q\times 1} \end{cases}$$

Completing with non essential variables (see step 2 in section 4) leads to the definition of factor $\overline{\psi}_{24}$ with non essential variables z_1, z_3 which can be written in TT format as follows ($\times$ is the matrices product)

$$\overline{\psi}_{24}(z_1, z_2, z_3, z_4) = G_{24}^{(1)}[z_1] \times G_{24}^{(2)}[z_2] \times G_{24}^{(3)}[z_3] \times G_{24}^{(4)}[z_4]$$

with, $\forall\, z_1, z_2, z_3, z_4$:

$$\begin{cases} G_{24}^{(1)}[z_1] = 1 & \in \mathbb{R}^{1\times 1} \\ G_{24}^{(2)}[z_2] = M_{24}[z_2] & \in \mathbb{R}^{1\times Q} \\ G_{24}^{(3)}[z_3] = \mathbb{I}_{\mathrm{Q}} & \in \mathbb{R}^{Q\times Q} \\ G_{24}^{(4)}[z_4] = V_{24}[z_4] & \in \mathbb{R}^{Q\times 1} \end{cases}$$

which can be sketched as

$\overline{\psi}_{24}(z_1, z_2, z_3, z_4) =$ 

1	$M_{24}[z_2]$	$\mathbb{I}_q$	$V_{24}[z_4]$
1×1	$1 \times Q$	$Q \times Q$	$Q \times 1$

The product of those matrices $(1,1)$, $(1,Q)$, (Q,Q) and $(Q,1)$ is a real. This is variable separation for the factor $\overline{\psi}_{24}(z_1, z_2, z_3, z_4)$.

This can be generalized to any n by computing all $G_{ij}^{(k)}[z_k]$ for $1 \le k \le n$ as

$$\left|\begin{array}{lll} k < i & \Rightarrow G_{ij}^{(k)}[z_k] = 1 & \in \mathbb{R}^{1\times 1} \\ k = i & \Rightarrow G_{ij}^{(k)}[z_i] = M_{ij}[z_i] & \in \mathbb{R}^{1\times Q} \\ i < k < j & \Rightarrow G_{ij}^{(k)}[z_k] = \mathbb{I}_{\mathrm{Q,Q}} & \in \mathbb{R}^{Q\times Q} \\ k = j & \Rightarrow G_{ij}^{(k)}[z_j] = V_{ij}[z_j] & \in \mathbb{R}^{Q\times 1} \\ k > j & \Rightarrow G_{ij}^{(k)}[z_k] = 1 & \in \mathbb{R}^{1\times 1} \end{array}\right. \tag{C14}$$

Such a calculation is done for any pair $(i,j) \in E$. We then have

$$\forall\, 1 \le i < j \le n, \quad \forall\, 1 \le k \le n, \qquad \overline{\psi}_{ij}(z_1, \ldots, z_n) = \prod_{k=1}^{n} G_{ij}^{(k)}[z_k]$$

where the product is the matrices product. This is separation of variables for factor $\overline{\psi}_{ij}$.

Computation of the matrices $A_k(z_k)$**.**

Here, we can borrow the same paths as in Novikov et al. (2014). In our toy example, it leads to

$$A_k(z_k) = G_{12}^{(k)}(z_k) \otimes G_{13}^{(k)}(z_k) \otimes G_{14}^{(k)}(z_k) \otimes G_{23}^{(k)}(z_k) \otimes G_{24}^{(k)}(z_k) \otimes G_{34}^{(k)}(z_k).$$

Here, $A_k(z_k)$ is the Kronecker product of matrices which depend each on z_k. Then it is a matrix which depends on z_k. This can be generalised as

$$A_k(z_k) = \bigotimes_{1 \leq i < j \leq n} G_{ij}^{(k)}(z_k). \tag{C15}$$

We then have

$$\psi(z_1, \ldots, z_n) = \prod_{k=1}^{n} A_k(z_k),$$

where the terms $A_k(z_k)$ are matrices and $\prod$ is the matrices product. This is variable separation for the non normalised joint distribution ψ. Let us note that the TT-rank of TT-matrix $A_k(z_k)$ is one, which is a useful property for deriving the fusion of cores in the calculation. To show this, let us write $A_k(z_k)$ in TT-matrix format. Therefore, let $G_{ij}^{(k)}(z_k)$ be with coefficients $G_{ij}^{(k)}(z_k)\left[\alpha_{ij}^{(k)}, \beta_{ij}^{(k)}\right]$ with $1 \leq \alpha_{ij}^{(k)}, \beta_{ij}^{(k)} \leq r$. Then, from the definition of the Kronecker product,

$$A_k(z_k)\left[\left(\alpha_{12}^{(k)}, \ldots, \alpha_{n-1,n}^{(k)}\right); \left(\beta_{12}^{(k)}, \ldots, \beta_{n-1,n}^{(k)}\right)\right] = \prod_{1 \leq i < j \leq n} G_{ij}^{(k)}(z_k)\left[\alpha_{ij}^{(k)}, \beta_{ij}^{(k)}\right]$$

which can be written

$$A_k(z_k)\left[\left(\alpha_{12}^{(k)}, \ldots, \alpha_{n-1,n}^{(k)}\right); \left(\beta_{12}^{(k)}, \ldots, \beta_{n-1,n}^{(k)}\right)\right] = \prod_{1 \leq i < j \leq n} G_{ij}^{(k),z_k}\left(\alpha_{ij}^{(k)}, \beta_{ij}^{(k)}\right)$$

by reordering the upper and lower indices, and shows that the TT-rank is one as a product of scalars. It can be seen on a simpler example: Let $A = G_1 \otimes G_2$ be a matrix, with $G_1 := G_1[i_1, j_1]$ and $G_2 := G_2[i_2, j_2]$ with coefficients. Then, by definition of Kronecker product, A can be expressed as a TT-matrix of TT-rank one, by $A[(i_1, i_2); (j_1, j_2)] = G_1[i_1, j_1]\, G_2[i_2, j_2] = g_1(i_1, j_1)\, g_2(i_2, j_2)$ where g_1, g_2 are scalars. As the TT-rank of $A_k(z_k)$ is one, the TT-rank of TT-matrix $B_k = \sum_{z_k} A_k(z_k)$ is Q.

Appendix D A situation where binary marginals are product of unary ones.

Here we show that, given two variables Z_i and Z_j, taking value in $\{1, \ldots, Q\}$, such that there exists q with $\mathbb{P}(Z_i = q) = 1$, then, the binary marginals $\mathbb{P}(Z_i, Z_j)$ are the products of the unary marginals. Let us have $Q = 3$. Binary marginals can be organized as elements in a 3×3 matrix. The unary marginals are the row-wise or

column-wise sums. It can be displayed as

$$\begin{array}{ccc|c} a & b & c & x \\ a' & b' & c' & x' \\ a'' & b'' & c'' & x'' \\ \hline y & y' & y'' & 1 \end{array}$$

with

$$\begin{cases} a+b+c & = x \\ a'+b'+c' & = x' \\ a''+b''+c'' & = x'' \\ a+a'+a'' & = y \\ b+b'+b'' & = y' \\ c+c'+c'' & = y'' \end{cases}$$

and all terms and marginals in this matrix being in $[0, 1]$. Let us assume without loss of generality that $x = 1$. So, $x' = x'' = 0$. From $a', b', c' \geq 0$ and $a' + b' + c' = x' = 0$, it follows that $a' = b' = c' = 0$, and the same for a'', b'', c''. So, the matrix is

$$\begin{array}{ccc|c} a & b & c & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline y & y' & y'' & 1 \end{array}$$

We can observe that each term in the matrix is the product of the marginals of its row and column. For example, $a = y \times 1$ as $y = a + 0 + 0 = a$. Note that is it not required that y, or y' or y'' is 1 or 0. This demonstration can be extended to $Q > 3$.

References

Abouabdallah, M.A., N. Peyrard, and A. Franc. 2022. Does clustering of DNA barcodes agree with botanical classification directly at high taxonomic levels? Trees in french guiana as a case study. *Molecular Ecology Resources 22*(5): 1746–1761. https://doi.org/https://doi.org/10.1111/1755-0998.13579 .

Al Daas, H., G. Ballard, and P. Benner. 2022. Parallel algorithms for tensor-train arithmetic. *SIAM Journal of Scientific Computing* 44: C25–C53. https://doi.org/10.1137/20M1387158 .

Al Daas, H., G. Ballard, P. Cazeaux, E. Hallman, A. Miedlar, M. Pasha, T.W. Reid, and A.K. Saibaba. 2023. Randomized algorithms for rounding in the Tensor-Train format. *SIAM Journal of Scientific Computing* 45: A74–195. https://doi.org/10.1137/21M1451191 .

Ansari, M., A. Ahmad, S. Khan, G. Bhushan, and M. Siddique. 2020. Spatiotemporal clustering: a review. *Artificial Intelligence Review 53*(4): 2381–2423. https://doi.org/10.1007/s10462-019-09736-1 .

Barbillon, P., S. Donnet, E. Lazega, and A. Bar-Hen. 2017. Stochastic block models for multiplex networks : An application to a multilevel network of researchers. *Journal*

of the Royal Statistical Society: Series A Statistics in Society 180(1): 295–314. https://doi.org/10.1111/rssa.12193 .

Carroll, J.D. and J.J. Chang. 1970. Analysis of individual differences in multidimensional scaling via $n-$ways generalization of eckart-young decomposition. *Psychometrika* 35: 283–319 .

Casella, G. and E.I. George. 1992. Explaining the Gibbs Sampler. *The American Statistician 46*(3): 167–174 .

Coppi, R. and S. Bolasco. 1989. *Multiway data analysis.* Amsterdam: Elsevier.

Daudin, J.J., F. Picard, and S. Robin. 2008. A mixture model for random graph. *Statistics and Computing 18*(2): 173–183. https://doi.org/10.1007/s11222-007-9046-7 .

de Lathauwer, L., B. de Moor, and J. Vandewalle. 2000. A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications 21*(4): 1253 .

de Silva, V. and L.H. Lim. 2008. Tensor rank and the ill posedness of the best low rank approximation, problem. *SIAM Journal on Matrix Analysis and Applications 30*(3): 1084–1027 .

Dempster, A., N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39: 1–22 .

Deza, M.M. and E. Deza. 2016. *Encyclopedia of Distances* (Fourth ed.). Springer.

Ducamp, G., P. Bonnard, A. Nouy, and P.H. Wuillemin 2020. An efficient low-rank tensors representation for algorithms in complex probabilistic graphical models. In *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, Skørping, Denmark.

Fannes, M., B. Nachtergaele, and R.F. Werner. 1992. Finitely Correlated States on Quantum Spin Chains. *Communications in Mathematical Physics* 144: 443–490 .

Faskowitz, J., X. Yan, X.N. Zuo, and O. Sporns. 2018. Weighted stochastic block models of the human connectome across the life span. *Scientific Reports* 8 .

Funke, T. and T. Becker. 2019. Stochastic block models: A comparison of variants and inference methods. *PLOS ONE 14*(4): e0215296. https://doi.org/10.1371/journal.pone.0215296 .

Geman, S. and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*(6): 721–741. https://doi.org/10.1109/TPAMI.1984.4767596 .

Gusfield, D. 1997. *Algorithms on strings, trees, and sequences.* Cambridge, UK: Cambridge University Press.

Harshman, R.A. 1970. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics* 16: 1–84 .

Holland, P.W., K.B. Laskey, and S. Leinhardt. 1983. Stochastic blockmodels: First steps. *Social Networks 5*(2): 109–137. https://doi.org/https://doi.org/10.1016/0378-8733(83)90021-7 .

Horn, R.A. and C.R. Johnson. 2012. *Matrix Analysis* (Second ed.). Cambridge.

Häggström, O. 2002. *Finite Markov Chains and Algorithmic Applications.* . Cambridge University Press. (London Mathematical Society Student Texts, Series Number 52).

Kalmbach, P., A. Blenk, M. Kluegel, and W. Kellerer 2017. Generating Synthetic Internet- and IP-Topologies using the Stochastic-Block-Model. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management*, Lisbon, Portugal.

Kolda, T.G. and B.W. Bader. 2009. Tensor decomposition and applications. *SIAM Review 51*(3): 455–500 .

Koller, D. and N. Friedman. 2009. *Probabilistic Graphical Models - Principles and Techniques.* MIT Press.

Kroonenberg, P.M. and J. de Leeuw. 1980. Principal Component Analysis of three modes data by means of Alternating Least Square algorithms. *Psychometrika 45*(1): 69–97 .

Lyu, Z., D. Xia, and Y. Zhang. 2023. Latent space model for higher-order networks and generalized tensor decomposition. *Journal of Computational and Graphical Statistics 32*(4): 1–17. https://doi.org/10.1080/10618600.2022.2164289 .

Mariadassou, M., S. Robin, and C. Vacher. 2010. Uncovering latent structure in valued graphs: A variational approach. *The Annals of Applied Statistics 4*(2): 715 – 742. https://doi.org/10.1214/10-AOAS361 .

Miele, V. and C. Matias. 2017. Revealing the hidden structure of dynamic ecological networks. *Royal Society Open Science 4*(6) .

Murphy, K.P. 2012. *Machine Learning: A Probabilistic Perspective.* MIT Press.

Novikov, A., A. Rodomanov, A. Osokin, and D. Vetrov 2014. Putting MRFs on a Tensor Train. In *Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014.*

Orus, R. 2014. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. *Annals of Physics* 349: 117–158 .

Oseledets, I., E. Tyrtyshnikov, and N. Zamarashkin. 2011. Tensor-Train Ranks for Matrices and Their Inverses. *Computational Methods in Applied Mathematics 11*(3): 394–403 .

Oseledets, I.V. 2009. A new tensor decomposition. *Doklady Mathematics 80*(1): 495–496 .

Oseledets, I.V. 2011. Tensor-Train decomposition. *SIAM Journal of Scientific Computing* 33: 2295–2317 .

Oseledets, I.V. and S.V. Dolgov. 2012. Solution of linear systems and matrix inversion in the tt-format. *SIAM Journal of Scientific Computing 34*(34): A2718–A2739 .

Peyrard, N., M.J. Cros, S. de Givry, A. Franc, S. Robin, R. Sabbadin, T. Schiex, and M. Vignes. 2019. Exact or approximate inference in graphical models: why the choice is dictated by the treewidth, and how variable elimination can be exploited. *Australian and New Zealand Journal of Statistics 61*(2): 89–133. https://doi.org/doi:10.1111/anzs.12257 .

Robert, C. and G. Casella. 2004. *Monte Carlo statistical methods (second edition)*. Springer Verlag.

Savicky, P. and J. Vomlel. 2007. Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetica 43*(5): 747–764 .

Smith, P.D. and M.S. Waterman. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology* 147: 195–197 .

Strang, G. 2019. *Linear Algebra and Learning from Data*. Wellesley Cambridge Press.

Tucker, L.R. 1966. Some mathematical notes on three-modes factor analysis. *Psychometrika 31*(3): 279–311 .

Vidal, G. 2003. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Physical Review Letters* 91. https://doi.org/DOI:10.1103/PhysRevLett.91.147902 .

Wainwright, M.J. and M.I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* 1: 1–305 .

Wrigley, A., W.S. Lee, and N. Ye 2017, 06–11 Aug. Tensor belief propagation. In D. Precup and Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning*, Volume 70, pp. 3771–3779. PMLR.

Computing SBM marginals with Tensor-Train decomposition: supplementary material

Mohamed Anwar Abouabdallah[1,2], Olivier Coulaud[3], Nathalie Peyrard[4*], Alain Franc[1,2]

[1]Université de Bordeaux, INRAE, UMR BIOGECO, Cestas, 33612, France.
[2]Pleiade, EPC INRIA-INRAE, Université de Bordeaux, Talence, 33405, France.
[3]Concace, EPII INRIA, Talence, 33405, France.
[4*]Université de Toulouse, INRAE, UR MIAT, Castanet-Tolosan, 31320, France.

*Corresponding author(s). E-mail(s): nathalie.peyrard@inrae.fr;
Contributing authors: maabouabdallah@gmail.com;
olivier.coulaud@inria.fr; alain.franc@inrae.fr;

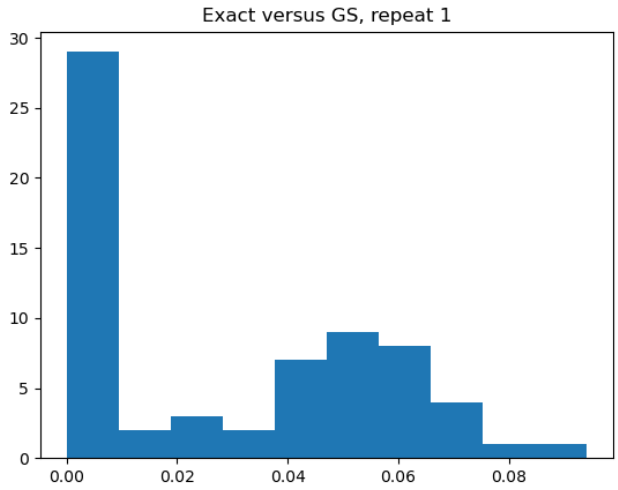


Fig. 1 Example of an histogram of the absolute difference between the value of $PSC_{i,j}$ computed with the exact method and with GS, for each pair (i, j) of variables.