

The interplay between ranking and communities in networks

Laura Iacovissi^{1,2,*} and Caterina De Bacco^{1,†}

¹Max-Planck Institute for Intelligent Systems, Cyber Valley, Tuebingen 72076, Germany

²Bosch Industry on Campus Lab, University of Tuebingen

Community detection and hierarchy extraction are usually thought of as separate inference tasks on networks. Considering only one of the two when studying real-world data can be an oversimplification. In this work, we present a generative model based on an interplay between community and hierarchical structures. It assumes that each node has a preference in the interaction mechanism and nodes with the same preference are more likely to interact, while heterogeneous interactions are still allowed. The algorithmic implementation is efficient, as it exploits the sparsity of network datasets. We demonstrate our method on synthetic and real-world data and compare performance with two standard approaches for community detection and ranking extraction. We find that the algorithm accurately retrieves each node’s preference in different scenarios and we show that it can distinguish small subsets of nodes that behave differently than the majority. As a consequence, the model can recognise whether a network has an overall preferred interaction mechanism. This is relevant in situations where there is no clear “a priori” information about what structure explains the observed network datasets well. Our model allows practitioners to learn this automatically from the data.

I. Introduction

In interacting systems, an observed tie between two individuals can often be explained by the existence of groups or a hierarchical organization. For instance, in social networks interactions between people can be explained by the communities the individuals belong to. In animal interaction networks, animals attack each other strategically based on some underlying notion of ranking perceived between them to determine their dominance in a hierarchy. When modelling network datasets, one typically observes only the set of interactions, but communities and ranking are hidden variables that need to be learned from the data, problems referred to as community detection and ranking extraction. Often, practitioners consider only one of the two for the dataset at hand, expecting a clear contribution for either community structure or hierarchy in determining edge formation. However, there can be situations where this distinction is blurry, and it is not clear what among these two effects plays a bigger role in explaining the data. For instance, students may report friendship relationships based on the groups they belong to or based on some hidden notion of hierarchy between them, thus reporting what friendship they aspire to have instead. The problem is that in these two cases the input dataset, a directed and possibly weighted network, looks the same. It is a list of edges $i \rightarrow j$ and their weight w , but one may not know what mechanism best explains the observed data. Unless a practitioner has a strong *a priori* expectation about how the system works, it is not clear how to distinguish if the network was generated by community structure or by hierarchical organization. The question thus is how to learn this interplay between communities and hierarchies from the data in an automatic way.

A large variety of algorithms are available for extracting communities and ranking from networks, spanning from simple heuristics and deterministic approaches to probabilistic ones. Standard algorithms for ranking entities are based on spectral methods, e.g. PageRank [1] and Eigenvector [2]. They are based on random walks on network and output real-valued scores. A different family of approaches considers ordinal rankings. These are typically extracted by finding an optimal permutation of the nodes that minimizes some penalty function. Relevant examples are Minimum Violation Rank [3, 4], SerialRank [5] and SyncRank [6]. Other approaches that consider real-valued scores extracted from pairwise preferences are those based on Random Utility Models [7], such as the Bradley-Terry-Luce (BTL) model [8, 9]. A different approach is that of SPRINGRANK [10], a physically-inspired ranking algorithm that computes real-valued scores via minimizing the energy of a system of springs representing the directed observed interactions. In terms of community detection models, there are various traditional approaches like graph partitioning, spectral clustering modularity-based algorithms or divisive algorithms [11]. In this work we focus on those based on probabilistic generative models [12–15] like the Stochastic Block Model (SBM) [16] and its variants. These have several advantages, including the possibility of sampling synthetic networks with a given community structure and predicting missing links. Most importantly, they allow for a probabilistic treatment, which is the approach we adopt here to tackle our problem.

* laura.iacovissi@uni-tuebingen.de

† caterina.debacco@tuebingen.mpg.de

In most cases, these algorithms are applied independently, i.e. one either extracts communities or ranking, and the practitioners decides a priori which model is most appropriate. There have been some attempts to consider both communities and ranking as hidden variables on networks, assuming some underlying interaction between the two mechanisms. For instance, Chen et al. [17] combine clustering and ranking by first inferring non-overlapping groups using a variant of SBM and then retrieving the within-cluster popularity of nodes of different types within each group, which in turns influences the community they belong to. Here the assumption is that nodes belong to groups and there is a ranking of nodes within each group. A recent work has studied the ranking communities problem, addressing the detection of communities by ranking them using information flow techniques [18]. An additional way of mixing the concepts of community detection and hierarchy is hierarchical clustering. The main idea is that there exists a hierarchy of communities that can be organized via a tree structure [19, 20]. Finally, we mention clustering algorithms used for ranking data, i.e. data that are intrinsically embedded with a hierarchical structure given as metadata [21]. All of these settings are fundamentally different from the problem considered here, in that they assume an intrinsic rank of communities or an intrinsic clustering of ranks. In this manuscript, instead, we assume that nodes interact mainly either because of community affinity or ranking, and we want our algorithm to learn the preferred mechanisms of each node automatically from the data. This is a relevant problem in networks where the two mechanisms coexist. For example, an individual can have a contact with another individual because of homophily or because of prestige. The former case can occur when there are some common attributes and preferences, so that individuals recognize them as part of the same market. The latter corresponds to cases in which the individuals are perceiving themselves as part of some league in terms of prestige, and they aim at connecting with someone in the same league or slightly above [22]. In these systems, models considering just one mechanism will be likely to recognize a subset of the interactions as noisy observations, or even interpret them in their own terms, leading to a distorted interpretation of the underlying patterns.

To address this problem, in this work we propose a probabilistic model capable of recognizing the community and ranking structures in a network with coexisting mechanisms and quantifying the extent to which each individual prefers one instead of the other. Our model considers latent variables encoding the division of nodes in clusters, the hierarchical organization of nodes and how every node prefers to interact. The generative model as we define it, allows us to address also the problems of predicting missing links in the data and assigning a preferred interaction type to each node, i.e. community affinity or ranking similarity. We validate our model on synthetic data and showcase its applications on two real datasets where the impact of community and ranking mechanisms differ. We find that it is capable of correctly assigning to each node its preferred interaction mechanism (hierarchy or community) with high confidence, i.e. all the probabilities are close to zero or one. In addition, also the coefficient representing the overall preferred mechanism is always close to the ground-truth value. All of this is achieved without losing accuracy on the edge prediction task, whose performances on the limit cases are close to the ones of the baseline methods.

II. Generating networks with coexisting community and hierarchical structures

In this work, we are interested in modelling networks with underlying coexisting community and hierarchical structures. We refer to the network mainly through its matrix representation, i.e. the adjacency matrix $A = \{A_{ij}\}_{i,j=1}^N$, where N is the number of nodes. The entry $A_{ij} \in \mathbb{N}$ represents the number of directed interactions $i \rightarrow j$ from node i to node j . Each interaction can be either due to affinity between nodes (community) or to competition between them (hierarchy). Examples are a node reporting another node as a friend (community) or a node endorsing another one based on some notion of prestige or strength (hierarchy). We are interested in scenarios where these two mechanisms coexist and the interaction type is not known in advance. The goal is thus to observe a network and distinguish edges based on which of these two mechanisms is more likely to explain the interaction. To this end, we assume nodes to be of two types: those that predominantly interact through community, and those that predominantly interact through hierarchy. The intuition is that nodes with the same preference are more likely to interact, in turns inducing the edge type (community or hierarchy). We further assume that the probability of a heterogeneous interaction between nodes of two different types is not null and can be considered as a third edge type. From a generative modelling perspective, this scenario can be understood as first drawing latent labels on nodes, corresponding to node type. Then drawing interactions between nodes from a specific distribution depending on their label, with its mean parameterized accordingly. Formally, the generative model is:

$$\sigma_i, \sigma_j \sim \text{Be}(\mu) \tag{1}$$

$$A_{ij} \sim \begin{cases} \text{Pois}(A_{ij}; S_{ij})^{\sigma_i} \text{Pois}(A_{ij}; M_{ij})^{1-\sigma_i} & \text{if } \sigma_i = \sigma_j \quad (\text{in-group interaction}) \\ \text{Pois}(A_{ij}; \delta_0) & \text{if } \sigma_i \neq \sigma_j \quad (\text{out-group interaction}) \end{cases}, \tag{2}$$

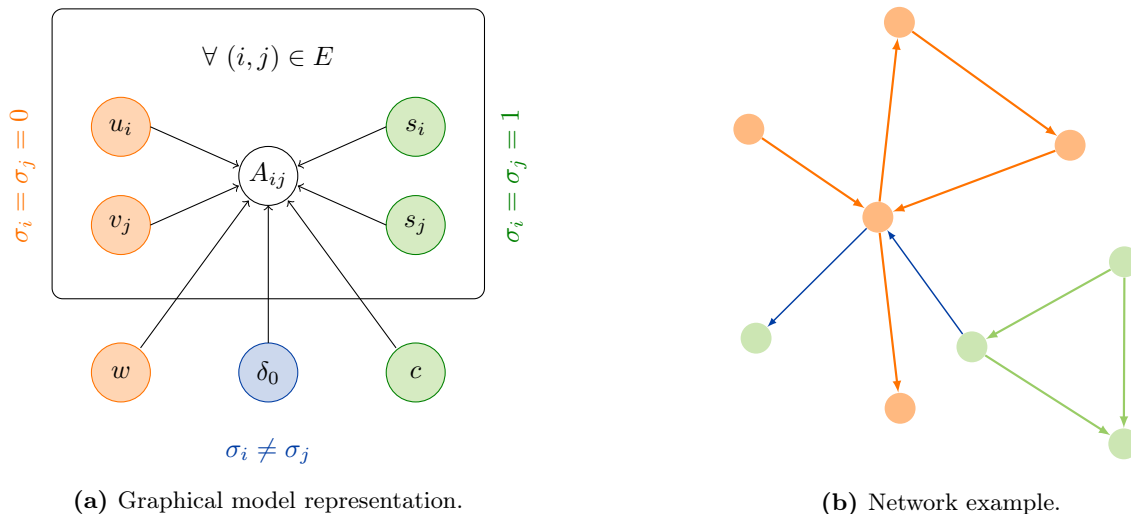


FIG. 1: Model visualization. (a) Graphical model: the entry of the adjacency matrix A_{ij} is determined by the community-related latent variables u, v, w (orange), by the ranking-related ones s, c (green) and by the out-group interaction parameter δ_0 (blue), depending on the values taken by the node type latent variables σ_i, σ_j . E denotes the set of network directed edges. (b) Example of possible realization of the model: orange nodes interact mainly via community, i.e. $\sigma_i = 0$, green ones via hierarchy, i.e. $\sigma_i = 1$. Orange and green edges are interactions between nodes of the same type (matching node color), while blue edges are interactions between nodes of different types.

where $\sigma_i \in \{0, 1\}$ represents the node type, $\mu \in [0, 1]$ its prior, and $\delta_0 \geq 0$ is a parameter that controls the density of edges between nodes of different type (typically small). The parameters M_{ij} and S_{ij} determine the community and hierarchy mechanisms, respectively. The procedure is repeated for each edge $i \rightarrow j$, since we assume conditional independence of the A_{ij} given the latent random variables. The model of Eqs. (1), (2) leads to the following network likelihood distribution

$$P(A|M, S, \sigma, \delta_0) = \prod_{ij} [\text{Pois}(A_{ij}; S_{ij})^{\sigma_i} \text{Pois}(A_{ij}; M_{ij})^{1-\sigma_i}]^{\delta_{\sigma_i \sigma_j}} [\text{Pois}(A_{ij}; \delta_0)]^{1-\delta_{\sigma_i \sigma_j}} . \quad (3)$$

Notice that with this parameterization we obtain that an edge type random variable $\delta_{\sigma_i \sigma_j} \in \{0, 1\}$ can be naturally defined in terms of σ using $\delta_{\sigma_i \sigma_j} = 2\sigma_i \sigma_j - \sigma_i - \sigma_j + 1$. It is Bernoulli distributed with parameter $\mu^2 + (1 - \mu)^2 = 1 - 2\mu(1 - \mu)$.

To model community interactions parametrized by M_{ij} we use MULTITENSOR [15] (MT), a mixed-membership variant of the SBM. Each node of the network belongs to a community to an extent represented by two membership vectors: $u_i = [u_{ik}]$ determines how much i belongs to the community k considering the amount of *out-going* edges; $v_i = [v_{ik}]$ only considers *in-coming* edges. An affinity matrix $w = [w_{kh}]$ encodes the density of edges between nodes in different communities. Note that all these quantities are positive but not necessarily normalized. These elements are combined in the expected number of community interactions as $M_{ij} = \sum_{k,h=1}^K u_{ik} v_{jh} w_{kh}$. This definition results in interactions more likely to exist between nodes with compatible community structure.

To model the hierarchical interactions parameterized by S_{ij} we use SPRINGRANK [10] (SR), a model that associates a score $s_i \in \mathbb{R}$ to each node and an interaction energy $-\frac{\beta}{2}(s_i - s_j - 1)^2$ to each edge $i \rightarrow j$ which regulates the probability of a hierarchical interaction as a Boltzmann weight. Here β is a hyperparameter that controls the strength of the hierarchy. These elements are combined in the expected number of hierarchical interactions as $S_{ij} = c \exp\left[-\frac{\beta}{2}(s_i - s_j - 1)^2\right]$, where c controls for network sparsity (see S1). This definition results in interactions more likely to exist between nodes with similar scores, i.e. close in rank.

We refer to our model as XOR. This is parameterized by $\theta = (u, v, w, s, c, \delta_0, \mu)$ and it models the network likelihood distribution $P(A|\theta, \sigma)$ as in Equation (3). Its computational complexity relies on the individual complexities of MULTITENSOR and SPRINGRANK, which are both optimized for sparse networks as it is often the case for real networks. Notice that for σ equal to a null or a unitary vector XOR reduces to MULTITENSOR or to SPRINGRANK, respectively.

III. Inference

Given a network adjacency matrix A , we want to infer the parameters θ and the node labels σ that best explain the observed data. To this end, we aim at maximizing $P(\theta|A) = \sum_{\sigma} P(\sigma, \theta|A)$, i.e. the maximum a posteriori estimate of θ . For convenience we maximize its logarithm instead, as the maxima coincide. We then take a variational approach by using Jensen's inequality

$$\log P(\theta|A) = \log \sum_{\sigma} P(\sigma, \theta|A) \geq \sum_{\sigma} q(\sigma) \log \frac{P(\sigma, \theta|A)}{q(\sigma)} =: \mathcal{L}(q, \theta), \quad (4)$$

where $q(\sigma)$ is a variational distribution over the node labels. This formulation of the problem turns it in a maximization of the function $\mathcal{L}(q, \theta)$ with respect to θ and q . In fact, since $q(\sigma)$ must sum to one, the exact equality between the second and third term in Equation (4) is achieved when $q(\sigma)$ is equal to the posterior $P(\sigma|\theta, A)$. However, this posterior may not be analytically accessible as the normalization is not tractable. In fact we have

$$P(\sigma|\theta, A) \propto P(\sigma, A|\theta) = \prod_i \mu^{\sigma_i} (1 - \mu)^{1 - \sigma_i} \prod_{ij} \text{Pois}(A_{ij}; S_{ij})^{\sigma_i \delta_{\sigma_i \sigma_j}} \text{Pois}(A_{ij}; M_{ij})^{(1 - \sigma_i) \delta_{\sigma_i \sigma_j}} \text{Pois}(A_{ij}; \delta_0)^{1 - \delta_{\sigma_i \sigma_j}}, \quad (5)$$

and this cannot be simply recast into a well-known probability distribution in σ (e.g. a fully factorized Bernoulli distribution). A careful reader will recognize that the $P(\sigma|\theta, A)$ is equivalent to an Ising model with unitary inverse temperature and Hamiltonian:

$$H_A(s|J, h) = \sum_{i,j} J_{ij} s_i s_j + \sum_i h_i s_i, \quad (6)$$

$$J_{ij} = \frac{\log \text{Pois}(A_{ij}; S_{ij}) + \log \text{Pois}(A_{ij}; M_{ij}) - 2 \log \text{Pois}(A_{ij}; \delta_0)}{4}, \quad (7)$$

$$h_i = \frac{1}{4} \sum_j (\log \text{Pois}(A_{ij}; S_{ij}) + \log \text{Pois}(A_{ji}; S_{ji}) - \log \text{Pois}(A_{ij}; M_{ij}) - \log \text{Pois}(A_{ji}; M_{ji})) + \frac{1}{2} (\log \mu - \log(1 - \mu)), \quad (8)$$

where here $s_i \in \{\pm 1\}$, $s(\sigma) = 2\sigma - 1$ and the couplings J are asymmetric, see S2.

To obtain a tractable expression for the variational distribution that estimates $P(\sigma|\theta, A)$, we use a mean-field approximation $q(\sigma) = \prod_i q_i(\sigma_i)$ assuming $q_i(\sigma_i) = \text{Be}(\sigma_i; Q_i)$. The goal is to find values of $\{Q_i\}_i$ such that the Kullback-Leibler divergence between the approximate posterior $q(\sigma)$ and the true posterior $P(\sigma|\theta, A)$ is minimized [23, 24]. Noting that the Hamiltonian in Eq. (6) corresponds to $-\log P(\sigma, A|\theta)$, the maximization of $\mathcal{L}(q, \theta)$ with respect q is equivalent to minimize a variational free energy $F(q, \theta)$ defined as:

$$F(q, \theta) = \sum_{\sigma} q(\sigma) H_A(s(\sigma), |J(\theta), h(\theta)) - S(q) = -\mathcal{L}(q, \theta) + \log P(\theta), \quad (9)$$

with the first term being the internal energy of q and S the entropy function of the product of Bernoulli distributions $q(\sigma)$.

By performing this minimization for q we obtain that the optimal parameters that are included in Algorithm 1, i.e. Equations (11) to (13), see S3 for detailed derivations. This result can also be obtained using the standard self-consistency equation for an Ising model $\sigma_i = \tanh\left(h_i + \sum_j J_{ij} \sigma_j\right)$ using J_{ij} and h_i as in Equations (7) and (8). One can in principle use alternative approximations more complex than mean-field [23], for instance the Bethe approximation, at the cost of increasing computational complexity. This is left for future work.

The values of Q_i are also point-estimates for the variables σ_i , as for a Bernoulli distribution $\mathbb{E}_q[\sigma_i] = Q_i$. Differentiating $\mathcal{L}(q, \theta)$ with respect to θ and setting this to zero gives the updates for the parameters θ . The full derivation is reported in S4, while the results can be seen in Algorithm 1 where we show the overall EM algorithmic routine. This algorithm does not guarantee convergence to the global maximum of the variational log-likelihood, but only to a local one. In practice, we perform different runs with different random initializations of the inputs and select the one with the best value of the $\mathcal{L}(q, \theta)$.

In the model we did not specify any prior for the θ . An alternative is to impose exponential priors for each u_{ik}, v_{ik} , independent and identically distributed with parameters λ_u, λ_v . This results in a L_1 -regularized $\mathcal{L}(q, \theta)$ that enforces sparse membership vectors and an overall contribution of the community mechanism close to $\|w\|_{max}$. To control the

growth of the value $\|w\|_{max}$ we can impose an exponential prior on w as well, parametrized by λ_w . The regularizer may prevent the SBM to overfit. This might be particularly relevant in the case of a block structure induced on the network by different leagues in the hierarchical organization of the nodes. The prior results in small modifications of the updates for the u, v, w , which are reported in equations (14) and (15). We do not set a prior on s since in previous studies [10] it was shown that adding a Gaussian prior may not necessarily lead to better prediction performance.

Algorithm 1 XOR : EM algorithm

Input: network $A = \{A_{ij}\}_{i,j=1}^N$; number of communities K ; inverse temperature β ; w 's, u 's and v 's a priori parameters $\lambda_u, \lambda_v, \lambda_w \in [0, 1]$.

Output: membership vectors $u = [u_{ik}]$, $v = [v_{ik}]$; network-affinity matrix $w = [w_{kq}]$; ranking score vector $s = [s_i]$; outgroup interaction parameter δ_0 ; σ 's a priori parameter μ ; σ 's a posteriori parameters' vector $Q = [Q_i]$.

Initialize $u, v, w, s, c, \delta_0, \mu, Q$ at random.

Repeat until \mathcal{L} converges:

1. Calculate ρ and Q (E-step):

$$\rho_{ijkh} = \frac{u_{ik}v_{jh}w_{kh}}{\sum_{lm} u_{il}v_{jm}w_{lm}}, \quad (10)$$

$$Q_i = \frac{f_{i1}}{f_{i1} + f_{i2}}, \quad (11)$$

$$f_{i1} = \mu \prod_{j \neq i} [\text{Pois}(A_{ij}; S_{ij})\text{Pois}(A_{ji}; S_{ji})]^{Q_j^{\text{old}}} [\text{Pois}(A_{ij}; \lambda_0)\text{Pois}(A_{ji}; \lambda_0)]^{(1-2Q_j^{\text{old}})}, \quad (12)$$

$$f_{i2} = (1 - \mu) \prod_{j \neq i} [\text{Pois}(A_{ij}; M_{ij})\text{Pois}(A_{ji}; M_{ji})]^{Q_j^{\text{old}} - 1}; \quad (13)$$

2. Update parameters θ (M-step):

- i) for each node i and community k update memberships:

$$u_{ik} = \frac{\sum_{jh} (Q_i Q_j - Q_i - Q_j + 1) A_{ij} \rho_{ijkh}}{\lambda_u + \sum_{jh} (Q_i Q_j - Q_i - Q_j + 1) v_{jh} w_{kh}}, \quad (14)$$

- ii) for each pair (k, h) update affinity matrix:

$$w_{kh} = \frac{\sum_{ij} (Q_i Q_j - Q_i - Q_j + 1) A_{ij} \rho_{ijkh}}{\lambda_w + \sum_{ij} (Q_i Q_j - Q_i - Q_j + 1) u_{ik} v_{jh}}; \quad (15)$$

- iii) update the sparsity coefficient and the ranking scores for each node i :

$$c = \frac{\sum_{ij} Q_i Q_j A_{ij}}{\sum_{ij} Q_i Q_j \exp\left[-\frac{\beta}{2}(s_i - s_j - 1)^2\right]}, \quad s_i = \frac{\sum_j Q_i Q_j s_j [A_{ji} + A_{ij}] + Q_i Q_j [A_{ij} - A_{ji}]}{\sum_j Q_i Q_j [A_{ji} + A_{ij}]}; \quad (16)$$

- iv) update outgroup interaction parameter:

$$\delta_0 = \frac{\sum_{ij} A_{ij} (2Q_i Q_j - Q_i - Q_j)}{\sum_{ij} (2Q_i Q_j - Q_i - Q_j)}. \quad (17)$$

- v) update μ parameter:

$$\mu = \frac{\sum_i Q_i}{N}. \quad (18)$$

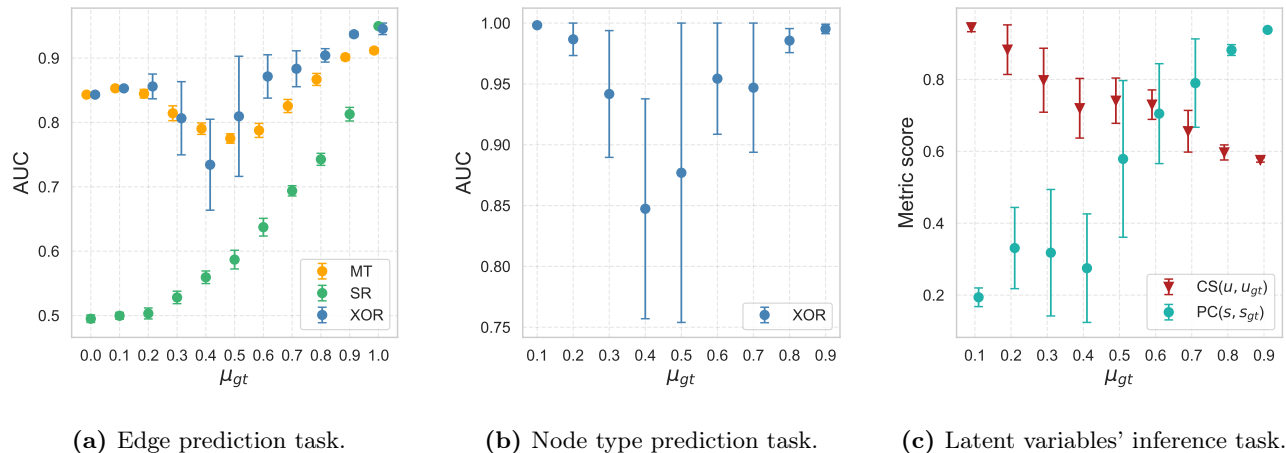


FIG. 2: Performances on synthetic networks for different tasks. The mean value across folds is reported, indicating also the standard deviations. For the edge prediction task, our model is compared with the baseline methods MULTITENSOR and SPRINGRANK. We vary the proportion of expected node with $\sigma_i = 1$, i.e. preferring a hierarchy-based interaction. Only the not regularised version of XOR is shown since it is the most effective in this scenario.

IV. Results

A. Results on synthetic data

The XOR model outputs the parameters related to the community (u, v, w) , the hierarchical structure (s, c) and the node types σ from the observed network data A . When the ground-truth values of θ and σ are available, we can measure the performance of the model in recovering the community structure, the ranking of nodes and their type. For these three tasks we consider as performance metrics the cosine similarity (CS), the Pearson's correlation (PC) and Area Under the Curve (AUC), respectively, between the ground-truth and the inferred values. In the absence of ground-truth, we can indirectly evaluate the model fitness via edge prediction tasks in cross-validation settings where we hide a subset of the matrix A (test set), fit the model in the remaining subset (training set) and test the ability to predict the missing edges (test set).

We validate the model on synthetic data generated using the XOR generative model with $N = 500$, average degree $\langle k \rangle = 20$, $\beta = 5$ and varying the ground-truth value of $\mu_{gt} \in [0, 1]$. Specifically, we generate networks with $K = 3$ communities of equal-size unmixed group membership, a hierarchy with $l = 3$ leagues, i.e. the scores $\{s_i\}_{i=1}^N$ are drawn from a mixture of Gaussians with means $\{-4, 0, 4\}$ and standard deviations $\{1, 0.5, 1\}$; we set $\delta_0 = 0.01$. We draw five different independent samples for each set of parameters. The inference algorithm is tested by using 5-fold cross-validation for splitting the data into train and test sets, and ran with five different random initializations on each graph instance. We set the hyperparameters K and β equal to the ground-truth values, while we use grid search for selecting the best value of the regularization $\lambda = \lambda_v = \lambda_u = \lambda_w \times 0.1$.

We find that XOR predicts missing edges robustly and consistently across different values of μ_{gt} and better than baseline models that consider only community structure (MT) or only hierarchical structure (SR), see Figure 2 (left). The performance is not monotonic in μ_{gt} : we obtain high values of AUC when μ_{gt} is close to 0 or when μ_{gt} is close to 1. These are extreme scenarios where a large majority of the nodes are predominantly interacting either via communities or hierarchy. As one of these two mechanisms dominates, it is also easier to infer the parameters, hence the higher AUC values. The intermediate region $0.2 \leq \mu_{gt} \leq 0.8$ where the nodes distribute more evenly between the two mechanisms corresponds to cases in which inference is harder. Nevertheless, the model shows stable performance in this range, with AUC mean values never dropping below 0.7 and always comparable or better than MT, the best performing among the two baselines. A similar non-monotonic behaviour is observed for the node classification task, where we aim at predicting the node type σ using Q . While we observe a similar performance drop in the same intermediate regime, performance is robust, as the average AUC is always higher than 0.85. Finally, the performance in recovering communities is good in the regime that most favours community structure ($\mu_{gt} < 0.4$) while recovering of the hierarchy is poor, and vice-versa in the opposite regime ($\mu_{gt} > 0.6$). This is intuitive, as when most of the nodes predominantly interact via communities, their score is irrelevant, and therefore cannot be recovered well. What matters is the ability to recover the structure corresponding to the main mechanism at play, i.e. high cosine similarity when $\mu_{gt} < 0.5$ or high Pearson's coefficient when $\mu_{gt} > 0.6$. We find that XOR performs well in this task, with a

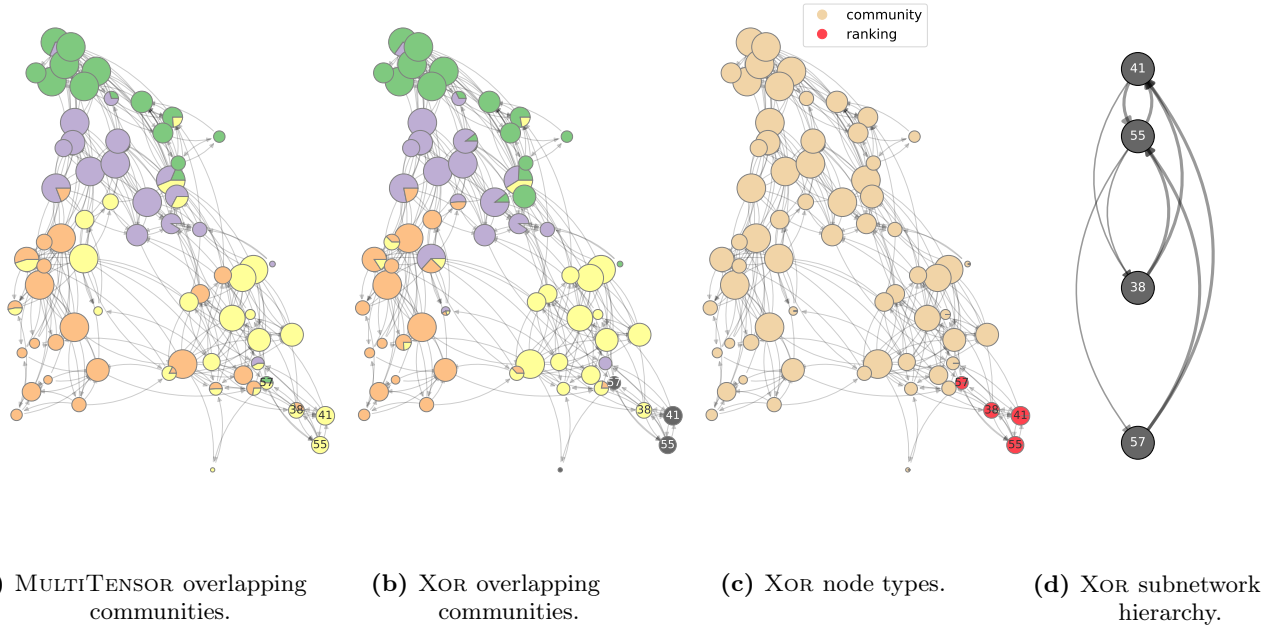


FIG. 3: Application on High School network: (a)-(b) comparison between the communities detected by MULTITENSOR and XOR, (c) node types Q visualization and (d) hierarchy in the subnetwork of nodes preferring the competitive mechanism. Nodes' positions are assigned: (a)-(c) using a spring layout and (d) using the scores s inferred by the XOR algorithm. Communities are selected by normalizing the v membership vector (similar results are obtained with u) and colors of the pie markers are assigned according to the community (mixed) membership in each group. The dark grey nodes have null membership vector. Both the algorithms select $K = 4$ as optimal number of communities, as output using a five-fold cross-validation scheme and grid search.

boost in performance in recovering the ranking to values above 0.7 in the regime where hierarchy dominates. Similarly, cosine similarity increases above 0.7 when communities dominate. These results suggest that the model is not only able to predict missing edges and nodes' type, but also to distinguish the node-level latent features that determine how nodes interact, regardless of their type (community membership or scores).

B. Results on real data

Application on high school network

As a first example of application of this model, we consider a dataset of a network of high school students [25]. This describes the perceived interactions of a group of 67 high school students. Each student is being asked “who are you friend of” in the fall of 1957 and the spring of 1958, and the answers are aggregated on the same edge allowing weights with values 1 or 2. Agreement in the response is not ensured, hence the network is directed. It is reasonable to expect that students belong to groups (communities) and this influences the answers they give. However, a fraction of the students may not belong to any group and instead nominate others based on their perceived ranking of the students. For instance, one may nominate whom they aspire to befriend. Figure 3 shows what happens when we apply XOR to this dataset. The figure shows the estimates of node type σ_i , describing the preferred behavior of each student. As we can see, most of the students have a $\sigma_1 \approx 0$, their nominations follow a community structure. However, we obtain four individuals with high σ_i , meaning that their preferences are mainly based on ranking. Even though they have a degree similar to that of other students, they are not well connected with the rest of the network as they mostly interact among themselves (there are only 5 other students that nominate one of them as friends). Comparing the reciprocity coefficient for the whole set of students with the one for the subgraph interacting mainly via hierarchy, we find that it increases from a value of 0.51 to a 0.88. In fact, this smaller network is missing only three edges for being a (directed) clique. Considering the direction and weight of the edges of the subgraph made of these four individuals, the ranking is meaningful as it reveals insights on the group social dynamics, as shown in Figure 3(d), where the

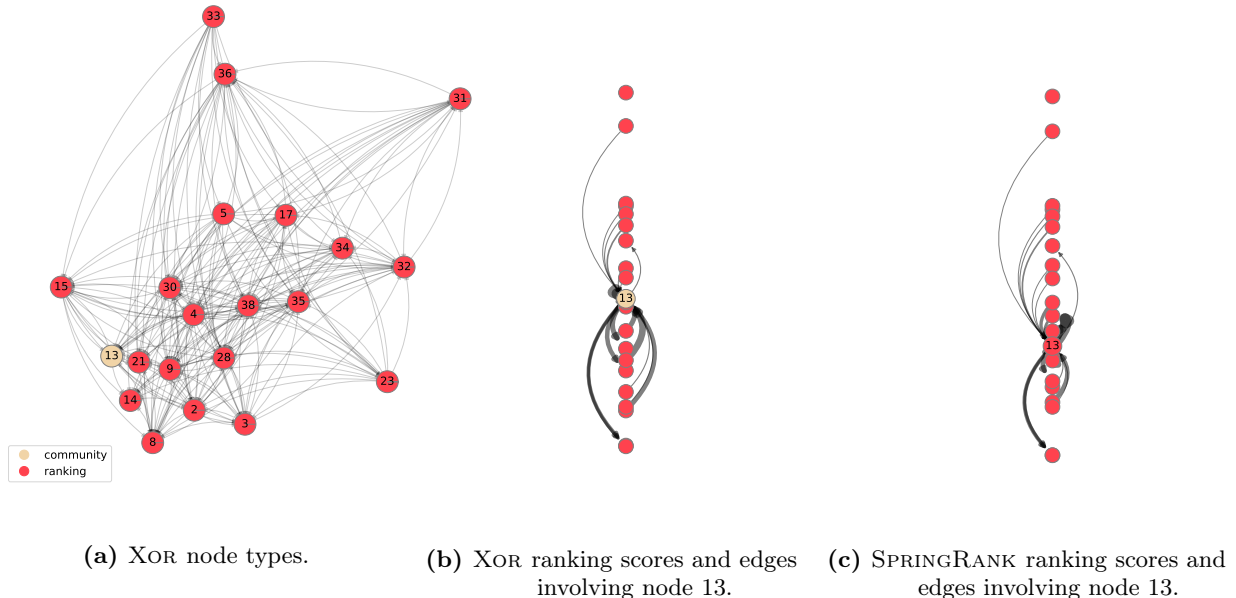


FIG. 4: Application on network of parakeets, group 1: visualization of (a) the node types Q ; (b) ranking scores, as inferred by the XOR model; (c) ranking scores inferred by the SPRINGRANK model. In (b) and (c) we highlight the interactions involving node 13, the only one that XOR infers as interacting via communities, i.e. $\sigma_{13} = 0$. The node positions are (a) determined with force atlas and (b)-(c) based on the inferred score s by XOR and SPRINGRANK respectively. Both XOR and MULTITENSOR detect no meaningful community structure, since they select $K = 1$ from five-fold cross validation and grid search.

hierarchy highlights consistency between inferred score (the position of the nodes) and the weight and direction of the interactions. Namely, the individual with highest ranking in that subset ($i = 41$), is also the individual that makes fewer friendship nominations, while the others tend to nominate them more often.

In the same figure, we show the community structure inferred by our model and that using MULTITENSOR, which considers only this mechanism for edge generation. As we can see, XOR outputs slightly different communities, as the yellow and blue nodes are partially mixed in the two cases. What is interesting, is that most of the nodes that have $\sigma_i = 1$ are not assigned to any community: again, we observe that the latent variable related to the mechanism not used for interacting is meaningless. Our models thus were able to extract a subgraph where hierarchy structure could meaningfully explain the directed interactions within that subgraph and also distinguish this from the remaining part of the networks where community memberships had likewise a meaningful interpretation. This showcases how practitioners should consider σ_i together with μ_i and s_i to fully characterize nodes.

Application on parakeets network

A second case study is the application to a network of directed aggressions among captive monk parakeets (*Myiopitta monachus*) [26]. Each directed edge contained the number of aggressive attacks between parakeets in two study groups, the first made of 21 individuals (G1), the second of 19 (G2). Each group was created and then observed for 24 days, divided into four 6-day study quarters. Insights from behavioural ecology suggest that the patterns of aggression correlate with an underlying dominance hierarchy: parakeets direct their aggression strategically, aiming at improving their position in the hierarchy [27]. Hence, we expect them to have a prevalent hierarchical latent structure, i.e. we expect $\sigma_i = 1$ for most of the individuals.

We performed experiments on the two groups, both extracting a single network from each quarter and aggregating on the quarters. The results on aggregated and not aggregated versions are similar for both groups, apart from more noisy results on the first quarter of both G1 and G2 given by a hierarchy not mature enough for being clearly detected [27]. The resulting inference on the aggregated G1 group is shown in Figure 4(a): only one node is predicted to use community-based interactions. This is also reinforced by the results of cross-validation tests to extract the number of communities – the value $K = 1$ achieves the best AUC score – and by inspecting u, v , which have mainly

null entries. We can deduct that this node is an anomaly in the social group, who interact with the other individuals with a random behaviour, rather than a strategic one. Hence, its score is to be considered irrelevant, which is in accordance with the fact that it is placed in a middle-ranking position while having a high number of incoming connections (33 incoming vs 40 outgoing, considering the weights), see Figure 4(b). Note that the results achieved by SPRINGRANK on the same network are similar: in Figure 4(c) $i = 13$ is placed 13th, close to the 10th position assigned by XOR, while the interaction pattern is not well in agreement with that. Again, it is behaving as an anomaly, but the SPRINGRANK algorithm cannot learn it as it is not designed to distinguish node types.

V. Discussion

The XOR model captures coexisting hierarchical and community mechanisms in networks. Being a generative model, it can be used for producing synthetic benchmarks with the desired level of interplay between the two mechanisms. It relies on a principled mathematical formulation with interpretable latent variables and its algorithmic implementation is optimized for sparse systems. In particular, it allows for automatic extraction of main patterns of interactions involving subsets of nodes. We gave examples of this by considering networks of friendship nominations among high school students and of aggressive interactions between monk parakeets. In the case of friendship nominations, XOR highlighted a small subnetwork of four individuals whose interactions stood out from the crowd. Similarly, for the aggression in parakeets, it spotted an individual outlier whose interactions do not seem to align with those observed involving other individuals in the group.

We considered here an efficient, but possibly limited, mean-field approximation to perform parameters' inference. Its connections with well-known models from statistical physics suggest as natural direction for future developments that of deploying more complex approximations, e.g. using belief propagation [28]. While we expect this to lead to more accurate approximate posterior distributions, this may come at the price of increasing complexity, we leave this as an open problem for future work. From a modelling perspective, an interesting direction for future work is to explore different ways of modelling interaction preferences. Here we assigned a latent variable σ to each node, but it would be interesting to investigate how results change when considering latent variables on edges instead. This choice may be more natural in scenarios where individuals form ties on a case-by-case basis rather than predominantly via one of the two mechanisms explored here. This could potentially account for a further mechanism for edge formation, as reciprocity [29–31]. Similarly, when node attributes are available along with the network dataset, it would be compelling to adapt the model to suitably incorporate this extra information using insights from previous works [32].

In summary, in this work we make a first step to tackle problems with mixed underlying mechanisms determining edge formation in networks. While we showed examples of interesting patterns possibly arising as inferred by our model, we provide an open-source implementation of the code to facilitate future data explorations.

Data availability

The code used for the analysis and to generate the synthetic data is publicly available and can be found at <https://github.com/liacov/XOR-rankcom>.

Acknowledgements

L.I. work was fully supported by the Max Planck Institute for Intelligent Systems. Her current affiliation is Bosch Industry on Campus Lab, University of Tuebingen.

-
- [1] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web.*, Technical Report 1999-66 (Stanford InfoLab, 1999) previous number = SIDL-WP-1999-0120.
 - [2] P. Bonaacich, *American Journal of Sociology* **92**, 1170 (1987).
 - [3] I. Ali, W. D. Cook, and M. Kress, *Management Science* **32**, 660 (1986).
 - [4] M. Gupte, P. Shankar, J. Li, S. Muthukrishnan, and L. Iftode (2011) pp. 557–566.
 - [5] F. Fogel, A. d' Aspremont, and M. Vojnovic, in *Advances in Neural Information Processing Systems*, Vol. 27, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Curran Associates, Inc., 2014).
 - [6] M. Cucuringu, *IEEE Transactions on Network Science and Engineering* **3**, 58 (2016).
 - [7] K. E. Train, *Discrete choice methods with simulation* (Cambridge university press, 2009).

- [8] R. A. Bradley and M. E. Terry, *Biometrika* **39**, 324 (1952).
- [9] R. D. Luce, *Psychological review* **66**, 81 (1959).
- [10] C. De Bacco, D. B. Larremore, and C. Moore, *Science advances* **4**, eaar8260 (2018).
- [11] S. Fortunato, *Physics reports* **486**, 75 (2010).
- [12] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, *Foundations and Trends in Machine Learning* **2**, 129 (2010).
- [13] P. J. Bickel and A. Chen, *Proceedings of the National Academy of Sciences* **106**, 21068 (2009).
- [14] B. Ball, B. Karrer, and M. E. Newman, *Physical Review E* **84**, 036103 (2011).
- [15] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, *Physical Review E* **95**, 042317 (2017).
- [16] P. W. Holland, K. B. Laskey, and S. Leinhardt, *Social Networks* **5**, 109 (1983).
- [17] J. Chen, W. Dai, Y. Sun, and J. Dy, in *Proceedings of the 2015 SIAM International Conference on Data Mining* (SIAM, 2015) pp. 424–432.
- [18] M. Pirouz, in *Proceedings of the Future Technologies Conference (FTC) 2021, Volume 3*, edited by K. Arai (Springer International Publishing, Cham, 2022) pp. 292–300.
- [19] A. Clauset, C. Moore, and M. E. J. Newman, in *Statistical Network Analysis: Models, Issues, and New Directions*, edited by E. Airoldi, D. M. Blei, S. E. Fienberg, A. Goldenberg, E. P. Xing, and A. X. Zheng (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007) pp. 1–13.
- [20] M. T. Schaub and L. Peel, *CoRR abs/2009.07196* (2020), 2009.07196.
- [21] J. Jacques and C. Biernacki, *Journal of Statistical Planning and Inference* **149**, 201 (2014).
- [22] E. E. Bruch and M. E. J. Newman, *Science Advances* **4**, eaap9815 (2018), <https://www.science.org/doi/pdf/10.1126/sciadv.aap9815>.
- [23] M. Opper and D. Saad, *Advanced mean field methods: Theory and practice* (MIT press, 2001).
- [24] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, *Journal of the American statistical Association* **112**, 859 (2017).
- [25] J. Kunegis, in *Proc. Int. Conf. on World Wide Web Companion* (2013) pp. 1343–1350.
- [26] E. A. Hobson and S. DeDeo, “Data from: Social feedback and the emergence of rank in animal society,” <https://doi.org/10.5061/dryad.p56q7> (2016).
- [27] E. A. Hobson and S. DeDeo, *PLOS Computational Biology* **11**, 1 (2015).
- [28] M. Mezard and A. Montanari, *Information, physics, and computation* (Oxford University Press, 2009).
- [29] H. Safdari, M. Contisciani, and C. De Bacco, *Phys. Rev. Research* **3**, 023209 (2021).
- [30] H. Safdari, M. Contisciani, and C. D. Bacco, “Reciprocity, community detection, and link prediction in dynamic networks,” (2021), [arXiv:2112.09624 \[cs.SI\]](https://arxiv.org/abs/2112.09624).
- [31] M. Contisciani, H. Safdari, and C. D. Bacco, “Community detection and reciprocity in networks by jointly modeling pairs of edges,” (2021), [arXiv:2112.10436 \[cs.SI\]](https://arxiv.org/abs/2112.10436).
- [32] M. Contisciani, E. A. Power, and C. De Bacco, *Scientific reports* **10**, 1 (2020).

Supporting Information (SI)

S1. Controlling sparsity

In order to control the average degree value given by the ranking and community structures, we can introduce two parameters c_{SR}, c_{MT} s.t. $c_{SR} = c$ and $w_{kh} = c_{MT}\hat{w}_{kh}, \forall k, h$. We assume that the average degree is composed by three contributes, i.e.:

$$\langle k \rangle N = \langle k_{SR} \rangle N + \langle k_{MT} \rangle N + \varepsilon N = \mathbb{E} \left[\sum_{ij} A_{ij} \right], \quad (\text{S1})$$

with ε noise term given by the outgroup interactions. Under the XOR model, we have:

$$\mathbb{E} \left[\sum_{ij} A_{ij} \right] = \sum_{ij} \sum_{\sigma} P(\delta_{\sigma_i \sigma_j}) \mathbb{E}_{A|\sigma} [A_{ij}] = 2\mu(1-\mu)\delta_0 N^2 + (\mu^2 + (1-\mu)^2) \sum_{ij} \mu S_{ij} + (1-\mu) M_{ij} \quad (\text{S2})$$

Hence, imposing the equivalence for each term:

$$\varepsilon = 2\mu(1-\mu)\delta_0 N, \quad (\text{S3})$$

$$c_{SR} = \frac{\langle k_{SR} \rangle N}{\mu(\mu^2 + (1-\mu)^2) \sum_{ij} e^{-\beta H_{ij}}}, \quad (\text{S4})$$

$$c_{MT} = \frac{\langle k_{MT} \rangle N}{(1-\mu)(\mu^2 + (1-\mu)^2) \sum_{ijkh} u_{ik} v_{jh} \hat{w}_{kh}}. \quad (\text{S5})$$

Notice that this formulation assumes that the outgroup interaction parameter δ_0 has to respect the following bound:

$$\delta_0 \leq \frac{\langle k \rangle}{2\mu(1-\mu)N}. \quad (\text{S6})$$

S2. Mapping to an Ising model

In order to define this mapping, it is convenient to rewrite the posterior distribution of σ such that:

$$P(\sigma|A, \theta) \propto \exp(\log P(\sigma, A|\theta)) = \exp(-H_{\theta}(\sigma, A)) \quad (\text{S7})$$

for some energy function

$$H_{\theta}(\sigma, A) = - \sum_i h_i s_i(\sigma_i) - \sum_{ij} J_{ij} s_i(\sigma_i) s_j(\sigma_j), \quad 2\sigma_i - 1 = s_i(\sigma_i) \in \{-1, 1\}. \quad (\text{S8})$$

From now on, we will use $\bar{S}_{ij}, \bar{M}_{ij}, \bar{\lambda}_0$ to denote the logarithm of the poisson distributions in A_{ij} with means $S_{ij}, M_{ij}, \lambda_0$.

Considering the first summation appearing in $\log P(\sigma, A|\theta)$ (Eq.(5)), it is

$$\sum_i \sigma_i \log \mu + (1 - \sigma_i) \log(1 - \mu) = \sum_i \frac{s_i(\sigma_i) + 1}{2} \log \mu + \left(1 - \frac{s_i(\sigma_i) + 1}{2} \right) \log(1 - \mu) \quad (\text{S9})$$

$$= \sum_i \frac{1}{2} (\log \mu - \log(1 - \mu)) s_i(\sigma_i) + \text{const}, \quad (\text{S10})$$

where the constant is not relevant since will be discarded when normalizing the Boltzmann distribution. Hence we have

$$h_i^{(1)} \equiv \frac{1}{2} (\log \mu - \log(1 - \mu)) \quad (\text{S11})$$

as the first component of the h_i field.

To obtain the J_{ij} and $h_i^{(2)} = h_i - h_i^{(1)}$ fields, we consider the second summation appearing in $\log P(\sigma, A|\theta)$, which corresponds to $\log P(A|\sigma, \theta)$:

$$\log P(A|\sigma, \theta) = \sum_{ij} \sigma_i \delta_{\sigma_i \sigma_j} \bar{S}_{ij} + (1 - \sigma_i) \delta_{\sigma_i \sigma_j} \bar{M}_{ij} + (1 - \delta_{\sigma_i \sigma_j}) \bar{\lambda}_0 \quad (\text{S12})$$

$$= \sum_{ij} \left(\frac{s_i + 1}{2} \right) \left(\frac{s_i s_j + 1}{2} \right) \bar{S}_{ij} + \left(1 - \frac{s_i + 1}{2} \right) \left(\frac{s_i s_j + 1}{2} \right) \bar{M}_{ij} \quad (\text{S13})$$

$$+ \left(1 - \frac{s_i s_j + 1}{2} \right) \bar{\lambda}_0 \quad (\text{S14})$$

$$= \sum_{ij} s_i s_j \frac{\bar{S}_{ij} + \bar{M}_{ij} - 2\bar{\lambda}_0}{4} + \sum_i s_i \sum_j \frac{\bar{S}_{ij} + \bar{S}_{ji} - \bar{M}_{ij} - \bar{M}_{ji}}{4} \quad (\text{S15})$$

$$+ \frac{\lambda_0}{2} N^2 + \sum_{ij} \frac{\bar{S}_{ij} + \bar{M}_{ij}}{4} \quad (\text{S16})$$

Now we have:

$$J_{ij} = \frac{\bar{S}_{ij} + \bar{M}_{ij} - 2\bar{\lambda}_0}{4} \quad (\text{S17})$$

$$h_i^{(2)} = \frac{1}{4} \sum_j (\bar{S}_{ij} + \bar{S}_{ji} - \bar{M}_{ij} - \bar{M}_{ji}) \quad (\text{S18})$$

The final Hamiltonian is:

$$H(s|J, h) = \sum_{i,j} J_{ij} s_i s_j + \sum_i (h_i^{(1)} + h_i^{(2)}) s_i \quad (\text{S19})$$

where the J_{ij} are asymmetric.

S3. Mean-field approximation for q

We describe here the procedure follow for computing the approximation of the variational distribution q under mean-field assumption. We assume that we can factorize it as $q(\sigma) = \prod_i q_i(\sigma_i)$, $q_i(\sigma_i) = \text{Be}(Q_i)$. We can write our objective function as

$$\mathcal{L}(q, \theta) = Q_i \mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 1, \sigma_{j \neq i} | \theta)] + (1 - Q_i) \mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 0, \sigma_{j \neq i} | \theta)] + H_b(Q_i) + \text{const} . \quad (\text{S20})$$

Maximizing w.r.t. Q_i gives the equation:

$$\log \left(\frac{Q_i}{1 - Q_i} \right) = \mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 1, \sigma_{j \neq i} | \theta)] - \mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 0, \sigma_{j \neq i} | \theta)] , \quad (\text{S21})$$

which is solved for

$$Q_i = \frac{\exp(\mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 1, \sigma_{j \neq i} | \theta)] - \mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 0, \sigma_{j \neq i} | \theta)])}{1 + \exp(\mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 1, \sigma_{j \neq i} | \theta)] - \mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 0, \sigma_{j \neq i} | \theta)])} \quad (\text{S22})$$

$$= \frac{\exp(\mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 1, \sigma_{j \neq i} | \theta)])}{\exp(\mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 1, \sigma_{j \neq i} | \theta)] + \mathbb{E}_{j \neq i} [\log P(A, \sigma_i = 0, \sigma_{j \neq i} | \theta)])} \quad (\text{S23})$$

$$= \frac{\exp(\mathbb{E}_{j \neq i} [\log P(\sigma_i = 1 | A, \theta, \sigma_{j \neq i})])}{\exp(\mathbb{E}_{j \neq i} [\log P(\sigma_i = 1 | A, \theta, \sigma_{j \neq i})] + \mathbb{E}_{j \neq i} [\log P(\sigma_i = 0 | A, \theta, \sigma_{j \neq i})])} . \quad (\text{S24})$$

Since the formula for $P(A, \sigma | \theta)$ is known, we can compute this quantity and obtain $Q_i = q_i(\sigma_i = 1)$. Using the $\delta_{\sigma_i \sigma_k}$ parametrization, neglecting self-loops and discarding the terms not dependent on σ_i , we finally obtain the following

formula:

$$\rho_{ijkh} = \frac{u_{ik}v_{jh}w_{kh}}{\sum_{lm} u_{il}v_{jm}w_{lm}} , \quad (\text{S25})$$

$$Q_i = \frac{f_{i1}}{f_{i1} + f_{i2}} , \quad (\text{S26})$$

$$f_{i1} = \mu \prod_{j \neq i} [\text{Pois}(A_{ij}; S_{ij}) \text{Pois}(A_{ji}; S_{ji})]^{Q_j} [\text{Pois}(A_{ij}; \lambda_0) \text{Pois}(A_{ji}; \lambda_0)]^{(1-2Q_j)} , \quad (\text{S27})$$

$$f_{i2} = (1 - \mu) \prod_{j \neq i} [\text{Pois}(A_{ij}; M_{ij}) \text{Pois}(A_{ji}; M_{ji})]^{Q_j - 1} . \quad (\text{S28})$$

S4. \mathcal{L} maximization under mean-field assumption

Here we report the computations for defining the EM algorithm updates. The goal is to maximise the following quantity:

$$\mathcal{L}(q, \theta) = \sum_{\sigma} q(\sigma) \log \frac{P(\sigma, \theta | A)}{q(\sigma)} = \sum_{\sigma} q(\sigma) \log P(A | \sigma, \theta) + \sum_{\sigma} q(\sigma) \log P(\sigma | \mu) + H(q) \quad (\text{S29})$$

where H is the Entropy function. Considering the log-likelihood term, we can write it more explicitly as

$$\begin{aligned} \sum_{\sigma} q(\sigma) \log P(A | \sigma, \theta) = \sum_{i,j,\sigma} \left[\sigma_i q(\sigma) \delta_{\sigma_i \sigma_j} \log \text{Pois}(A_{ij}; S_{ij}) + (1 - \sigma_i) q(\sigma) \delta_{\sigma_i \sigma_j} \log \text{Pois}(A_{ij}; M_{ij}) \right. \\ \left. + q(\sigma) (1 - \delta_{\sigma_i \sigma_j}) \log \text{Pois}(A_{ij}; \delta_0) \right] . \end{aligned} \quad (\text{S30})$$

Consider the first term in Eq. (S30):

$$\sum_{i,j,\sigma} \sigma_i q(\sigma) \delta_{\sigma_i \sigma_j} \log \text{Pois}(A_{ij}; S_{ij}) = \sum_{ij} Y_{ij} \log \text{Pois}(A_{ij}; S_{ij}) ; \quad Y_{ij} := \sum_{\sigma} \sigma_i q(\sigma) \delta_{\sigma_i \sigma_j} = Q_i Q_j , \quad (\text{S31})$$

since under the MF assumption the following result holds:

$$\sum_{\sigma} \sigma_i q(\sigma) \delta_{\sigma_i \sigma_j} = \sum_{\sigma} q(\sigma) (2\sigma_i^2 \sigma_j - \sigma_i^2 - \sigma_i \sigma_j + \sigma_i) = 2\mathbb{E}[\sigma_i^2] \mathbb{E}[\sigma_j] - \mathbb{E}[\sigma_i^2] - \mathbb{E}[\sigma_i] \mathbb{E}[\sigma_j] + \mathbb{E}[\sigma_i] \quad (\text{S32})$$

$$= \mathbb{E}[\sigma_i] \mathbb{E}[\sigma_j] = Q_i Q_j . \quad (\text{S33})$$

Now consider the second term: the sum on σ can be rewritten as

$$\sum_{\sigma} (1 - \sigma_i) q(\sigma) \delta_{\sigma_i \sigma_j} = \sum_{\sigma} q(\sigma) \delta_{\sigma_i \sigma_j} - \sigma_i q(\sigma) \delta_{\sigma_i \sigma_j} = X_{ij} - Y_{ij} , \quad X_{ij} := \sum_{\sigma} q(\sigma) \delta_{\sigma_i \sigma_j} = 2Q_i Q_j - Q_i - Q_j + 1 , \quad (\text{S34})$$

obtained with a computation similar to the one in Eq. (S32).

Equation (S30) is then equal to

$$\sum_{\sigma} q(\sigma) \log P(A | \sigma, \theta) = \sum_{ij} \left[Y_{ij} \log \text{Pois}(A_{ij}; S_{ij}) + (X_{ij} - Y_{ij}) \log \text{Pois}(A_{ij}; M_{ij}) + (1 - X_{ij}) \log \text{Pois}(A_{ij}; \delta_0) \right] , \quad (\text{S35})$$

and the whole \mathcal{L} in Eq. (S29) now is

$$\mathcal{L}(q, \theta) = \sum_{ij} Y_{ij}(-S_{ij} + A_{ij} \log S_{ij}) + (X_{ij} - Y_{ij})(-M_{ij} + A_{ij} \log M_{ij}) + \sum_{ij} (1 - X_{ij})(-\delta_0 + A_{ij} \log \delta_0) \quad (\text{S36})$$

$$+ \sum_i Q_i \log(\mu) + (1 - Q_i) \log(1 - \mu) + H(q) \quad (\text{S37})$$

$$= \sum_{ij} Q_i Q_j (-S_{ij} + A_{ij} \log S_{ij}) + \sum_{ij} (Q_i Q_j - Q_i - Q_j + 1)(-M_{ij} + A_{ij} \log M_{ij}) \quad (\text{S38})$$

$$- \sum_{ij} (2Q_i Q_j - Q_i - Q_j)(-\delta_0 + A_{ij} \log \delta_0) + \sum_i Q_i \log(\mu) + (1 - Q_i) \log(1 - \mu) + \sum_i H_b(Q_i) . \quad (\text{S39})$$

Deriving the \mathcal{L} w.r.t. s_i , we have:

$$s_i = \frac{\sum_j Q_i Q_j s_j [A_{ji} + A_{ij}] + Q_i Q_j [A_{ij} - A_{ji}]}{\sum_j Q_i Q_j [A_{ji} + A_{ij}]} , \quad (\text{S40})$$

In order to have the equations for the community parameters we introduce a variational probability

$$p_{ijkh} = \frac{u_{ik} v_{jh} w_{kh}}{\sum_{kh} u_{ik} v_{jh} w_{kh}} . \quad (\text{S41})$$

that is representing the probability of observing an interaction between the nodes i, j because of their respective belonging to communities k, h .

Deriving the \mathcal{L} w.r.t. u_{ik} we obtain:

$$u_{ik} = \frac{\sum_{jh} (Q_i Q_j - Q_i - Q_j + 1) A_{ij} p_{ijkh}}{\sum_{jh} (Q_i Q_j - Q_i - Q_j + 1) v_{jh} w_{kh}} . \quad (\text{S42})$$

We find similar expression for v_{ih} and w_{kh} .

Similarly for μ :

$$\mu = \frac{1}{N} \sum_i Q_i . \quad (\text{S43})$$

For the sparsity c :

$$c = \frac{\sum_{ij} Q_i Q_j A_{ij}}{\sum_{ij} Q_i Q_j e^{-\frac{\beta}{2}(s_i - s_j - 1)^2}} . \quad (\text{S44})$$

For the noise mean δ_0 :

$$\delta_0 = \frac{\sum_{ij} A_{ij} (2Q_i Q_j - Q_i - Q_j)}{\sum_{ij} (2Q_i Q_j - Q_i - Q_j)} . \quad (\text{S45})$$