Irene Llobera Querol

Generalized Dot Product Models for Link Prediction

Degree Final Project

Supervised by Dr Marta Sales Pardo
and Dr Roger Guimerà Manrique

Bachelor's degree in Math and Physics Engineering

UNIVERSITAT ROVIRA i VIRGILI

Tarragona
2025

Abstract

Understanding the structure of complex networks is an important problem in network analysis, applicable to many disciplines that use graphs to model data. Two widely used models are the Random Dot Product (RDP) and the Mixed Membership Stochastic Block Model (MMSBM). While MMSBM is appreciated for its flexibility useful to model complex patterns, RDP is known for its simplicity and interpretability. However, RDP fails to model disassortative behavior in networks.

In this work we investigate the reason for this limitation and give empirical proof of it as well as of the generality of the MMSBM. We also propose a new generalization of the RDP using complex numbers. With these, our new method makes it possible to have negative inner products therefore helping the model capture disassortative structures. We compare the flexibility of the three models with synthetic and real-world netwoks proving that the RDP fails with disassortative structures while MMSBM and the new model are able to correctly model them.

Keywords: assortativity; disassortativity; block structure; Random Dot Product Graph, Mixed Membership Stochastic Block Model.

Resumen

Comprender la estructura de las redes complejas es un problema importante en el análisis de redes, aplicable a muchas disciplinas que usan grafos para modelar datos. Dos modelos ampliamente utilizados son el Random Dot Product (RDP) y el Mixed Membership Stochastic Block Model (MMSBM). Mientras que el MMSBM es apreciado por su flexibilidad para modelar patrones complejos, el RDP es conocido por su simplicidad e interpretabilidad. Sin embargo, el RDP no puede modelar comportamientos disasortativos en redes.

En este trabajo investigamos la razón de esta limitación y ofrecemos pruebas empíricas de ello, así como de la generalidad del MMSBM. También proponemos una nueva generalización del RDP utilizando números complejos. Con estos, nuestro nuevo método permite tener productos internos negativos, lo que ayuda al modelo a capturar estructuras disasortativas. Comparamos la flexibilidad de los tres modelos con redes sintéticas y reales, demostrando que el RDP falla con estructuras disasortativas, mientras que el MMSBM y el nuevo modelo son capaces de modelarlas correctamente.

Palabras clave: asortatividad; disasortatividad; estructura de bloques; Random Dot Product Graph, Mixed Membership Stochastic Block Model.

Resum

Entendre l'estructura de les xarxes complexes és un problema important en l'anàlisi de xarxes, aplicable a moltes disciplines que utilitzen grafs per modelar dades. Dos models àmpliament utilitzats són el Random Dot Product (RDP) i el Mixed Membership Stochastic Block Model (MMSBM). Mentre que el MMSBM és valorat per la seva flexibilitat per modelar patrons complexos, el RDP és conegut per la seva simplicitat i interpretabilitat. Tanmateix, el RDP no pot modelar comportaments disassortatius en xarxes.

En aquest treball investiguem la raó d'aquesta limitació i n'oferim proves empíriques, així com de la generalitat del MMSBM. També proposem una nova generalització del RDP utilitzant nombres complexos. Amb aquests, el nostre nou mètode permet tenir productes interns negatius, fet que ajuda el model a capturar estructures disassortatives. Comparem la flexibilitat dels tres models amb xarxes sintètiques i reals, demostrant que el RDP falla amb estructures disassortatives, mentre que el MMSBM i el nou model les poden modelar correctament.

Paraules clau: assortativitat; disassortativitat; estructura de blocs; Random Dot Product Graph, Mixed Membership Stochastic Block Model.

Acknowledgments

I would like to express my gratitude to Marta and Roger for all the help and guidance they have provided me in this project, from which I have learned a lot.

I want to thank all professors involved in GEMiF for their support and guidance these 4 years.

Finally, to my family and friends for their help and support.

Contents

1 Introduction

The study of complex networks has become a fundamental topic in modern data science, with applications in sociology, biology, neuroscience, and recommendation systems among others [1]. In many of these domains, it is observed that real-life networks often have an underlying structure that governs the observed connections between nodes. Understanding and modeling these structures is a key step in tasks such as link prediction, clustering, community detection, and understanding the dynamics of complex networks [1].

In many networks, the probability of a connection between two nodes depends on their attributes. These attributes, like what the node represents or the role it plays, can affect how likely it is to connect to other nodes. This creates patterns in the network that reflect how different types of nodes tend to interact. For example, in social networks, people with similar interests or backgrounds often connect with each other, which leads to assortative structures [2] [3]. In contrast, in biological or technological systems, it is more common for different types of nodes to connect rather than creating clusters of nodes with similar attributes. This results in disassortative patterns [3]. There are also many networks which exhibit a mix of both behaviours. Being able to capture these patterns is important for building accurate network models.

Among the most well-known models for inferring network structures are the Stochastic Block Model (SBM) and its generalization, the Mixed Membership Stochastic Block Model (MMSBM). These models use the concept of blocks or communities, where each node belongs to one or multiple of them. The behavior of each block is modeled and the interaction patterns among all nodes are determined [4]. This structure allows for the representation of both assortative and disassortative interactions, capturing different kinds of relationships both within and between communities. Thanks to this flexibility the MMSBM has proven especially powerful in modeling real-world networks with complex patterns [5].

The Random Dot Product (RDP) is also one of the most widely used models for graph inference for its simplicity and computational efficiency. This model assigns a vector to each node such that the probability of connection between any two nodes is the dot product of their assigned vectors [2]. While there are many advantages to this model, it has been observed that it doesn't correctly model all kinds of interactions. Particularly, it struggles when applied to networks with a disassortative structure. This limitation becomes significant when applying RDP to real-world networks that do not exhibit purely assortative patterns. Nevertheless, claims have been made, such as in Athreya, Fishkind, Tang, *et al.* [6], that the RDP is as general as the MMSBM and can model any network structure.

This work explores the strengths and limitations of the RDP when applied to networks of different structures and compares its effectiveness to the MMSBM. We aim to prove that the RDP has inherent constraints in its formulation that prevent it from correctly inferring disassortative networks. We prove this claim both theoretically and empirically, testing our models in synthetic and real-world networks.

Motivated by these limitations, we also aim to propose a new model that captures the simplicity of the RDP while making it general to all kinds of structures. Our proposed solution introduces complex numbers to the RDP. Using the increased flexibility offered by complex

numbers, including the possibility of negative inner products, our model is able to represent disassortative relationships while preserving the simplicity of the original RDP. By combining theory and experiments, we show that our method is a promising way to create models that are both flexible and easy to work with.

This work aims to contribute to the growing field of geometric graph inference by offering new ways to model complex networks with complex connection patterns. In particular, the improvements discussed can help capture structures that are often found in social networks, such as those related to inequality or group separation. Although analyzing these social dynamics is not the focus of the project, the methods we developed here may be useful to obtain a better understanding of such patterns in future applications.

1.1 Aims of the project

The objectives of this project are:

- Study the similarities and differences between RDPs and MMSBM's models theoretically.
- Study the limitations of the RDP mathematically.
- Propose a new model generalizing the RDP to make it applicable to networks with generalized connection patterns.
- Empirically compare the new model's performance to RDP and MMSBM in networks of different structures.
- Compare the performance of all three models in real-world networks.

2 Theoretical Background

In this section we examine different latent dimension inference models in their basic definitions and how they compare with each other.

2.1 Random Dot Product (RDP)

Random graph is a general term used to refer to graphs in which connections between vertices, the edges, are created based on a probability distribution. The classical example is the Erdos-Renyi model $G(n, p)$ in which a graph is generated with n nodes and each possible edge between two nodes exists with probability p independently from the other edges. This model provides a simple starting point for understanding how network systems behave when connections are made at random.

Kraetzl, Nickel and Scheinerman developed a new family of Random Graph models based on the dot product. In this model, each node i in the graph is assigned a random vector $v_i \in \mathbb{R}^d$ where each coordinate is independently and identically distributed over the real numbers. Each edge i, j exists with a probability $v_i \cdot v_j$ [2]. This framework introduces the idea of a latent space, specifically a subspace of $\mathbb{R}^d$, in which the underlying structure of the network is embedded.

As explained in Athreya, Fishkind, Tang, *et al.* [6], in a d-dimensional random dot product graph with n vertices, the latent positions of all vertices can be represented as a $n \times d$ matrix X. Therefore, since $\mathbb{P}[(v_i, v_j) \in E(G)] = X[i]X^\top[j]$ we can define a probability matrix P as $P = XX^\top$ where P_{ij} is the probability that the edge exists between v_i and v_j as defined before. The adjacency matrix A of a graph generated by P will consist of Bernoulli random variables with probability P_{ij}. Since we are dealing with undirected and unweighted graphs $A_{ii} = 0 \; \forall i$.

This model also highlights the role of latent dimensions. The dimension d of the latent space determines the complexity of the structure that the random graph can capture. A larger d allows for more intricate and flexible patterns of connectivity. Conversely, smaller values of d restrict the model to simpler forms, often resulting in less complex and more homogeneous graphs.

We can understand from the model definition that two nodes will behave identically if and only if they have the same vector representation. "Behaving the same" here means that, for any third node, they share the same probability of forming a connection—even if the actual, realized connections may differ due to the random nature of the process. Proof of this statement. The first implication:

$$X_i = X_j \quad \Rightarrow \quad \forall k \in 1, \ldots, n, \quad \mathbb{P}(A_{ik} = 1) = X_i^\top X_k = X_j^\top X_k = \mathbb{P}(A_{jk} = 1)$$

Second implication:

$$\forall k \in 1, \ldots, n, \quad \mathbb{P}(A_{ik} = 1) = \mathbb{P}(A_{jk} = 1) \quad \Rightarrow$$

$$X_i^\top X_k = X_j^\top X_k \quad \Rightarrow \quad (X_i - X_j)^\top X_k = 0 \quad \forall k$$

RDPG is not only a generative model for random graphs but also serves as a framework for inference. To do this, we must invert the process: starting from an observed graph and inferring the

latent vectors that likely generated it. This requires embedding techniques to map the nodes into a vectorial form in the latent space.

There are several approaches for embedding a graph into a latent space, each reflecting different assumptions about the structure of the network. A classical method is spectral embedding, in which the adjacency or Laplacian matrix of the graph is decomposed and the leading eigenvectors are used to define the node positions in a lower-dimensional space [6]. Another common approach is matrix factorization, where the adjacency matrix is approximated by the product of lower-rank matrices representing the latent features of the nodes. More recent developments include neural network-based methods, such as node2vec and graph neural networks, which learn node embeddings by optimizing representations that preserve structural patterns in the graph [1].

2.2 Stochastic Block Model (SBM)

A stochastic block model (SBM) is a statistical model used to describe network data that exhibits a block structure. In this context, the network's nodes are divided into groups, or blocks, and the probability of an edge between any two nodes depends only on the blocks to which they belong [7]. In the basic SBM one node can only belong to one group.

Let $G = (V, E)$ be a graph with $n = |V|$ nodes, where V is the set of nodes (or vertices) and $E \subseteq V \times V$ is the set of edges connecting pairs of nodes. Each node $i \in \{1, 2, ...n\}$ is assigned one of K groups with assignment function z. In this way $z(i)$ denotes the block to which node i belongs. Then we define the symmetric matrix P of dimensions $K \times K$ in which each P_{kl} denotes the probability of connection between nodes of group k and group l. Therefore, we define the probability of connection between any two pairs of nodes in those groups as:

$$\mathbb{P}(A_{ij} = 1 | z(i) = k, z(j) = l) = P_{kl} \tag{1}$$

The edges are assumed to be conditionally independent given the group assignments and connection probabilities between blocks. Therefore, the probability of the adjacency matrix A is:

$$\mathbb{P}(A \mid z, P) = \prod_{i<j} \mathbb{P}(A_{ij} \mid z(i), z(j)) = \prod_{i<j} P_{z(i)z(j)}^{A_{ij}} (1 - P_{z(i)z(j)})^{1-A_{ij}}. \tag{2}$$

2.2.1 Mixed Membership Stochastic Block Model (MMSBM)

The basic Stochastic Block Model (SBM) assumes that each node belongs to exactly one group or block. This hard assignment limits the model's expressiveness when dealing with networks that have complex community structures. In these cases, a way to increase the flexibility of the model is to increase the number of groups, which can quickly become inefficient and difficult to interpret.

The Mixed Membership Stochastic Block Model (MMSBM) is a more flexible approach to address these limitations. Instead of assigning each node to a single group, the MMSBM allows each

node to belong to multiple groups simultaneously [4]. This is achieved by associating each node to a membership vector whose entries indicate the degree to which a node belongs to each group. To find the probability of the existence of every edge, the model takes the degree of participation of each node to each group multiplied by the probability of connection between each pair of groups. This approach enables the MMSBM to capture more complex and realistic patterns in network data, such as overlapping communities [5].

Formally, we define the MMSBM as follows:

Given $G = (V, E)$ with $n = |V|$ nodes, each node $i \in \{1, 2, ...n\}$ is assigned a membership vector $\theta_i = (\theta_{i1}, \theta_{i2}, ...\theta_{iK})$ with $\sum_{k=1}^{K} \theta_{ik} = 1$. So that θ_{ik} can be interpreted as the percentage of node i belonging to group k. $P \in [0, 1]^{K \times K}$ is a symmetric matrix where P_{kl} gives the probability of an edge between a node in block k and a node in block l. The probability of connection between nodes i and j is calculated as:

$$\mathbb{P}(A_{ij} = 1 | \{\boldsymbol{\theta_i}\}, P) = \sum_{k=1}^{K} \sum_{l=1}^{K} \theta_{ik} \theta_{jl} P_{kl} \tag{3}$$

In Godoy-Lorite *et al.* [5] the model's parameters that optimize the likelihood are obtained through an expectation-maximization algorithm.

The algorithm starts by randomly initializing the group membership vectors $\{\boldsymbol{\theta}_i\}_{i=1}^{n}$, assigning uniform random values followed by normalization to ensure $\sum_k \theta_{ik} = 1$ for all i. The probability of connection matrix P is also initialized randomly, with values in $[0, 1]$, and symmetrized.

Expectation step: In this step we compute the auxiliary variable $\omega_{ij}(k, \ell)$, which can be interpreted as the posterior probability that the edge (i, j) is generated by node i belonging to group k and node j to group ℓ, given the current parameters:

$$\omega_{ij}(k, \ell) = \frac{\theta_{ik} \theta_{j\ell} \left[P_{k\ell} A_{ij} + (1 - P_{k\ell})(1 - A_{ij}) \right]}{\sum_{m=1}^{K} \sum_{q=1}^{K} \theta_{im} \theta_{jq} \left[P_{mq} A_{ij} + (1 - P_{mq})(1 - A_{ij}) \right]} \tag{4}$$

Maximization step: We update the parameters θ and P using the $\omega_{ij}(k, \ell)$ values computed in the E-step:

$$\theta_{ik}^{\text{new}} = \frac{1}{n} \sum_{j=1}^{n} \sum_{\ell=1}^{K} \omega_{ij}(k, \ell) \tag{5}$$

$$P_{k\ell}^{\text{new}} = \frac{\sum_{i,j} \omega_{ij}(k, \ell) \cdot A_{ij}}{\sum_{i,j} \omega_{ij}(k, \ell)} \tag{6}$$

These steps are iterated until convergence.

3 Results

3.1 Limitations of the RDP model

With the definitions of the models we notice that a key concept is that of similarity between nodes. In the RDP similar nodes, meaning nodes that behave similarly, have similar embeddings. In SBM and MMSB, similar nodes will belong to the same groups. In a general sense, the similarity between nodes can be classified into two different behaviors: assortativity and disassortativity.

Assortativity refers to the tendency of nodes with similar attributes or group memberships to be more likely to connect. This seems a natural behaviour in many real life scenarios: in social networks, people are more likely to form friendships with others who share similar interests, backgrounds, or demographic characteristics. We can also name this homophily, where edges are more common between similar nodes [3].

Disassortativity, on the other hand, describes the opposite pattern: a group of nodes all tend to connect to a different group of nodes but not with each other. It is common in networks where complementarity drives interactions [3]. In real life we might see this in a dating network, where people who share a ”type” will all behave similarly and want to date the same group of people. We can also name this heterophily.

In the SBM it is easy to see that an assortative graph will have high values in the diagonal of the P matrix, meaning that nodes are more likely to connect within the group. In contrast, disassortative graphs have higher values off the diagonal, meaning that nodes are more likely to connect between different groups than within the same group.

We are interested in finding whether our models do an equally good job at modeling both assortative and disassortative networks.to do this, it is important to understand how RDP and MMSBM relate to each other. As we have already mentioned, MMSBM can represent both assortativity and disassortativity as the probabilities of connections within groups are specifically stated for every pair of blocks. However, can the RDP generate disassortativity structures?

In Athreya, Fishkind, Tang, *et al.* [6] we find the claim: About the RDPG they state: *Our methodology is general enough to encompass networks that exhibit both homophily and heterophily*. Even further: *The mixed membership SBM allows for each vertex to be in a mixture of different blocks. Additionally, note that every RDPG is a MMSBM for some choice of K*.

This claim seems to point to the reasoning that the random dot product is equal to the probability calculation of the MMSBM. Notice, however, that while the RDP only involves the multiplication of the embedding vectors of each node, the product in MMSBM involves the probability matrix as well as the membership vectors.

$$\mathbb{P}(A_{ij} = 1|X) = X_i^\top X_j$$

$$\mathbb{P}(A_{ij} = 1|\{\boldsymbol{\theta_i}\}, P) = \sum_{k=1}^{K}\sum_{l=1}^{K} \theta_{ik}\theta_{jl}P_{kl},$$

If $P_{kl} = v_k^\top v_l$, then

$$\sum_{k=1}^{K}\sum_{l=1}^{K}\theta_{ik}\theta_{jl}P_{kl} = \sum_{k=1}^{K}\sum_{l=1}^{K}\theta_{ik}\theta_{jl}(v_k^\top v_l) = \left(\sum_{k=1}^{K}\theta_{ik}v_k\right)^\top \left(\sum_{l=1}^{K}\theta_{jl}v_l\right) = X_i^\top X_j$$

Meaning that any graph created or modeled by MMSBM could also be modeled by RDP.

However, is the assumption always true? Can we always decompose P as $P_{kl} = v_k^\top v_l$?

This has to do with the properties of the matrix, in particular, we can only decompose for **positive semidefinite matrices**. Meaning matrices that have no negative eigenvalues. How can we relate the concepts of disassortativity and positive semidefinite matrices?

The Generalized Sylvester's Criterion tells us that a matrix will be positive semidefinite if all its principal minors are non-negative. If we focus on 2×2 minors this links back to assortativity and disassortativity. In a network with some disassortative behaviour we should find some minors with higher values outside the diagonal that on the diagonal. This would indicate that two groups in the network tend to connect more to each other than they do among themselves. However a minor with higher values outside the diagonal that on the diagonal, accounting that all values are probabilities and therefore smaller than one, will always be negative. Therefore, a network with some disassortative behaviour will never be correctly modelled by RDP.

Even though we have proved that RDP cannot model disassortativity, Let's look into other claims by Athreya, Fishkind, Tang, *et al.* [6]. They claim: *The well-known stochastic blockmodel (SBM), in which each vertex belongs to one of K subsets known as blocks, with connection probabilities determined solely by block membership (Holland et al., 1983), can be represented as a random dot product graph in which all the vertices in a given block have the same latent positions.*

Let us look at why this is not true for disassortative graphs. Consider a disassortative graph with two groups, A and B: nodes in group A are more likely to connect to nodes in group B than to other nodes in A, and vice versa. Following the previous claim we assign identical embeddings u_A and u_B to all nodes in groups A and B, respectively. However, under a standard (Euclidean) dot product model where connection probabilities are given by $u_i^\top u_j$, conditions $u_A^\top u_B > u_A^\top u_A$ and $u_A^\top u_B > u_B^\top u_B$ cannot be satisfied simultaneously. This is because the dot product between two distinct vectors cannot exceed the squared norm of either vector unless they are nearly identical, which contradicts the requirement that $u_A \neq u_B$. $u_A \cdot u_B = |u_A||u_B| \cos\theta_{AB} > |u_A|^2$ and $u_A \cdot u_B > |u_B|^2$.

To capture disassortative interactions, the standard dot product model is not sufficient. A more flexible approach uses a generalised bilinear form $P_{ij} = v_i \Lambda v_j$, where Λ is a symmetric matrix that adjusts how different latent dimensions contribute to edge probabilities [6].

This formulation shares the same structure as the Mixed-Membership Stochastic Block Model and generalizes the Random Dot Product Graph. P and Λ have the same role. Because Λ is symmetric, it is always possible to rotate the vectors v_i so that it becomes diagonal. However, the basic RDP seems to assume that we can always reduce Λ to the identity matrix, which is not true. Furthermore, to model disassortativity, the diagonal Λ will include negative values, as they are

eigenvalues of a matrix that is not positive semidefinite. These negative entries allow the model to penalize similarity in certain latent directions, allowing the representation of disassortative mixing patterns. Without negative values in Λ, the dot product of two similar vectors will always be bigger than two very opposite vectors, as it happens in the basic RDP.

3.2 Complex model

Let us continue on the previous idea of including a matrix Λ in the dot product. We encountered the following problem: even though Λ can be diagonalized, if the diagonalized matrix contains negative values, it cannot be decomposed into a product of matrices. Therefore we can't use the basic RDP. This ties back to the condition of positive semidefiniteness discussed earlier: a matrix can be decomposed only if it is positive semidefinite, equivalently, if it has no negative eigenvalues. Since the values in a diagonal matrix are its eigenvalues Λ can never be decomposed for disassortative networks. However, the idea we introduce now is that it could be decomposed if the vectors allowed for complex numbers. Equivalently, to impose positive eigenvalues in Λ, the eigenvectors should have complex components.

A random dot product of complex numbers would essentially be a generalization of the basic RDP. As already discussed, a complex dot product will also model the graphs modelled by RDP but intuitively, it should also be able to model disassortative graphs. The complex values in the vectors will have the role of the negative values in diagonal Λ.

Formally, we define the new complex model as follows:

Let $G = (V, E)$ be a graph with $n = |V|$ nodes. $\vec{z_i}$ is the vector that represents node i. $z_{i\alpha} \in \mathbb{C}$ $\alpha = 1, ..., k$. For each vector the following is true:

$$\vec{z_i} \cdot \vec{z_i^*} = 1 \qquad \sum_\alpha z_{i\alpha} z_{i\alpha}^* = 1 \qquad z_{i\alpha} = r_{i\alpha} e^{i\phi_{i\alpha}}$$

The probability of existence of an edge is given by:

$$p(A_{ij} = 1 | \vec{z_i}, \vec{z_j}) = p_{ij} = (\vec{z_i} \cdot \vec{z_j}) (\vec{z_i} \cdot \vec{z_j})^* = \sum_{\alpha,\beta} z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^* \tag{7}$$

To simplify notation we define $R_{ij\alpha\beta} \equiv r_{i\alpha} r_{j\alpha} r_{i\beta} r_{j\beta}$ and $\Delta\Phi_{ij\alpha\beta} \equiv \phi_{i\alpha} + \phi_{j\alpha} - \phi_{i\beta} - \phi_{i\beta} = \phi_{i\alpha} + \Delta\Phi'_{ij\alpha\beta}$. Therefore:

$$p_{ij} = \sum_{\alpha\beta} R_{ij\alpha\beta} e^{i\Delta\Phi_{ij\alpha\beta}} \tag{8}$$

With the mathematical procedure detailed in Appendix A we finally get to the following equations for parameters $r_{i\alpha}$ and $\phi_{i\alpha}$

$$\tan \phi_{k\gamma} = -\frac{\sum_{j\neq k} \left(\frac{A_{kj}}{p_{kj}} - \frac{1-A_{kj}}{1-p_{kj}} \sum_\alpha R_{kj\gamma\alpha} \sin \Delta\Phi'_{kj\gamma\alpha} \right)}{\sum_{j\neq k} \left(\frac{A_{kj}}{p_{kj}} - \frac{1-A_{kj}}{1-p_{kj}} \sum_\alpha R_{kj\gamma\alpha} \cos \Delta\Phi'_{kj\gamma\alpha} \right)} \tag{9}$$

$$r_{kj}^2 = \frac{\sum_{j\neq k}[(A_{kj} - (1-A_{kj})\frac{p_{kj}}{1-p_{kj}})\sum_\alpha Re(\omega_{\alpha\gamma}(kj))]}{\sum_{j\neq k}[(A_{kj} - (1-A_{kj})\frac{p_{kj}}{1-p_{kj}})} \tag{10}$$

With auxiliary functions defined as:

$$\omega_{\alpha\beta}(ij) = \frac{R_{ij\alpha\beta}e^{i\Delta\Phi_{ij\alpha\beta}}}{p_{ij}} \tag{11}$$

$$\tilde{\omega}_{\alpha\beta}(ij) = \frac{\frac{1}{K^2} - R_{ij\alpha\beta}e^{i\Delta\Phi_{ij\alpha\beta}}}{1-p_{ij}} \tag{12}$$

We hoped that by implementing these equations, we could get the optimal complex vector solutions for each graph. However, during experimental validation, it was not possible to obtain solutions from them. This is probably due to a mistake or violation of a mathematical rule in the derivation. Since fixing this was not possible in time to include in this project, the experimental validation for the complex model was done using stochastic optimization, which randomly searches for values of the parameters that lower the loss, defined as follows:

$$\mathcal{L} = -\sum_{i\neq j} \log\left(A_{ij}\cdot P_{ij} + (1-A_{ij})\cdot(1-P_{ij})\right)$$

3.3 Experimental Validation

3.3.1 Method: Link prediction

The goal of the experimental validation is to show proof of the theoretical ideas presented so far by applying the different inference methods to various networks and comparing the results obtained to the expected. Our aim is to compare the RDP, the MMSBM and the new complex model explained in the previous section. For the RDP the inference method used has been the spectral embedding, implemented making use of the graspologic library available in Python. The inference for the MMSBM is done with equations 4, 5 and 6 explained before. Also, as mentioned earlier, the equations developed for the complex model did not provide coherent results, therefore, the inference was made using stochastic optimization, perturbing one vector at a time and keeping the changes that improve the value of the loss.

To be able to draw conclusions on the accuracy of each model, we need to establish a way to measure it. For the analysis of the synthetic networks we create 50 different graphs based on the same block structure. For each graph we apply increasing degrees of noise. For each degree of noise we take the corresponding percentage of all existing edges in the original network and move them to connect nodes that were not originally connected. Approximately, for degrees of noise above 0.3 the original structure of the network will already have degenerated greatly.

The comparison value that we will use is the AUC, Area Under the ROC Curve. It quantifies the probability that a model will assign a higher probability of existence to a link that does exist than

to a link that does not exist [8]. We take 10 true connections in the original matrix that have been erased by the introduction of noise, and 10 originally non-connected nodes. Using the probabilities inferred by each model, we rank the 20 candidate edges from highest to lowest probability of existence. A model with good predictive performance should place the erased edges near the top of the ranking. In contrast, a completely uninformative model will result in AUC values around 0.5, as it will rank edges randomly. In our analysis we calculate each AUC for each network and value of noise. We compute the average and the standard deviation and plot them to visually compare the results from the three methods.

It is important to understand that the performance of the model depends on the overall density of edges in the matrix. Even if some blocks have higher connection probabilities than others, a problem arises when those probabilities are still low, for example below 0.5. This means that even in the most densely connected block, any two nodes still have less than a 50% chance of being connected. As a result, many actual edges in that block may be missing in the original matrix. During the calculation of the AUC, a randomly sampled edge from this block is still likely to be absent. Even if the model correctly learns the block structure, it may assign high scores to these missing (but likely) edges, ranking them above observed zeros from less connected blocks, bringing the AUC score down.

3.3.2 Artificial networks

In order to give a first proof of concept, all the networks used in this section were created with a block model structure in which each node belonged fully to a group and groups had a probability of connecting with each other based on a probability matrix.

Starting with the most basic structure of assortativity and disassortativity, we created networks of 100 nodes divided into two blocs of equal size. We created an assortative network with probabilities:

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

and its disassortative counterpart:

$$P = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}$$

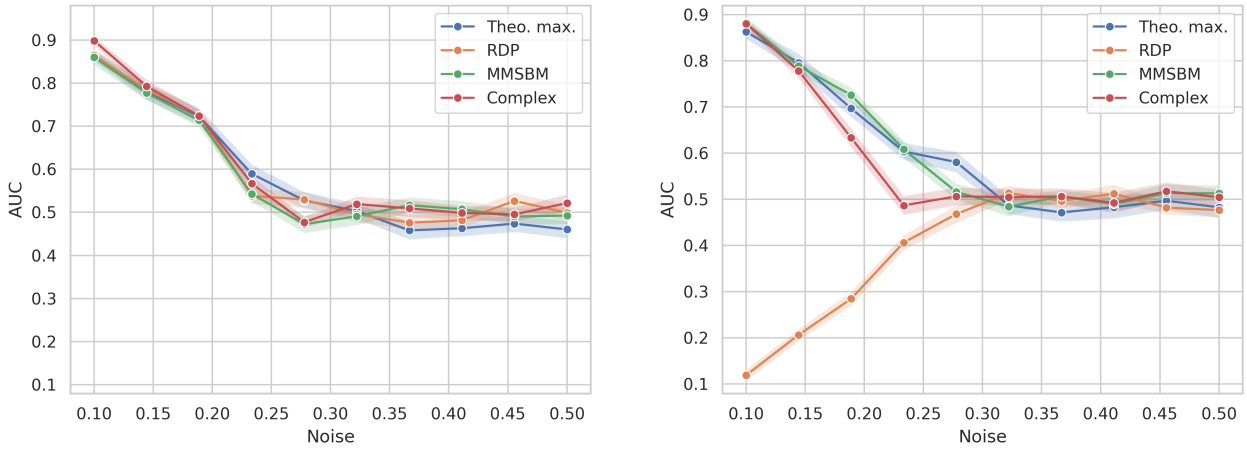
All methods were run with the same number of latent dimensions as the networks were originally generated. For the ones just explained $K = 2$. Since the inference of the MMSBM and the complex model depend on the initialization, for all networks and values of noise we have run the inference algorithm an average of 5 times and kept the result with the best log-likelihood and loss, respectively.

For all models, we expect to start with high values of AUC as the underlying structure will be very visible. At some point in the increase of noise the structure of the network will be so deteriorated that the inference methods will not be able to get any underlying structure and the models will be completely uninformative, leading to AUC values around 0.5. The plotted line labeled as ”Theoretical maximum” is the AUC values calculated with the probabilities that

generated the network. Even the original probabilities are not able to infer the structure of the network once the noise values become too high. We can't expect the models to perform better that the theoretical maximum.

Fig. 1 shows all three models perform similarly for the assortative version while on the disassortative graph we find that RDP goes to AUC values below 0.5. 0.5 would be the expected result if the inferred probabilities were not informative of the structure of the network since then the model would be just guessing. AUC values below 0.5 indicate that the model is ordering the edges in the AUC sample in the inverse. The model is inferring high probabilities where they should be low and low probabilities when they should be high. It is clear, therefore that RDP is "expecting" the network to be assortative and gives high probability of connecting to nodes of the same group and low probability to connections between groups.

In addition to that, we see that the complex model and MMSBM perform very similarly and also similar to the theoretical maximum. Both networks are able to extract the underlying group structure with very small differences between the assortative and disassortative case.



(a) Assortative

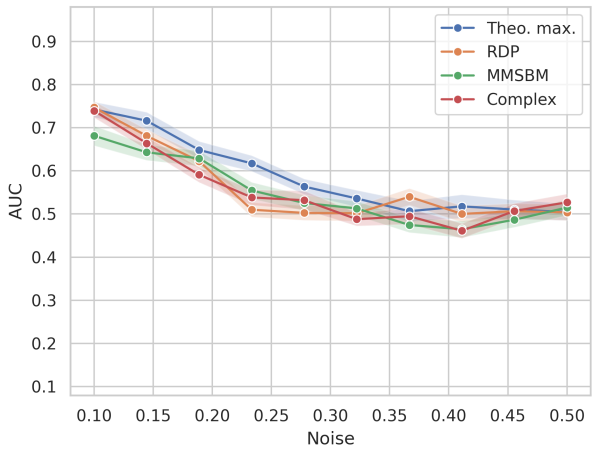
(b) Disassortative

Figure 1. Comparison in AUC among the different methods for 2 block networks generated with probabilities 0.9 and 0.1

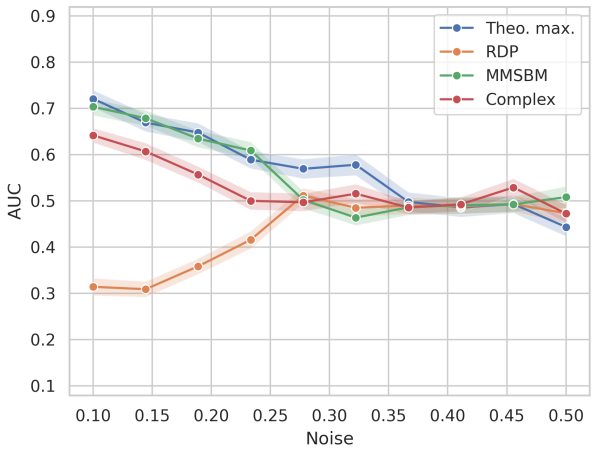
To see how the models behave in different cases, we generate networks with the same 2-block assortative and disassortative structure but this time with lower probabilities and, therefore, less dense. In the results in Fig. 2 the blocks with a higher probability of connecting have a 0.5 rate and the lower connected ones are kept at 0.1. The results are very similar to the previous networks. This time we do notice that the complex model got worse results than the MMSBM for the disassortative network. The RDP shows the same problem as before.

The AUC is generally lower for less dense matrices as even edges with the higher probability of connecting have a high chance of not connecting. In the AUC calculation these edges still count as "fake" and we expect the model to rank them lower than existing edges.

In Fig. 3 we used probabilities 0.2 and 0.05 in the network and we notice how all AUC results are lower. The three models perform very similarly for the assortative network but for



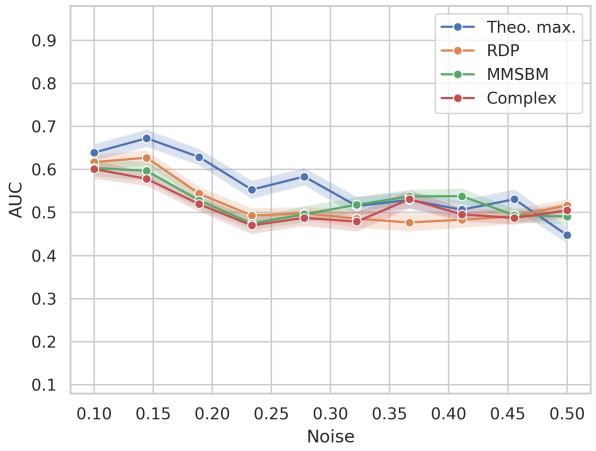
(a) Assortative



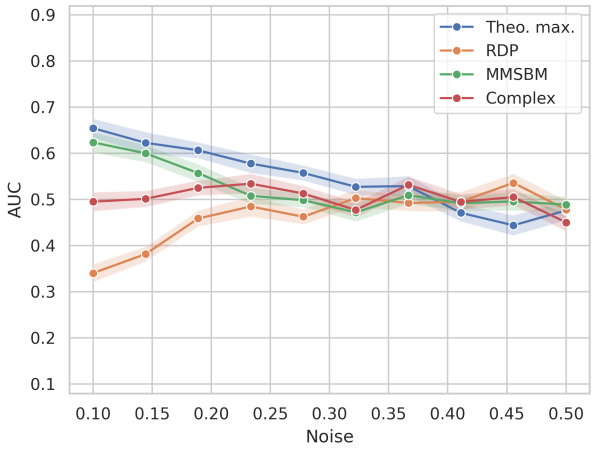
(b) Disassortative

Figure 2. Comparison in AUC among the different methods for 2 block networks generated with probabilities 0.5 and 0.1

the disassortative, we see how the complex model is stuck at AUC around 0.5, meaning it can't understand the structure of the network.



(a) Assortative



(b) Disassortative

Figure 3. Comparison in AUC among the different methods for 2 block networks generated with probabilities 0.2 and 0.05

Finally, we generate 2 networks with a structure of 3 and 4 blocks respectively. This allows us to introduce assortative and disassortative behaviour in the same network.

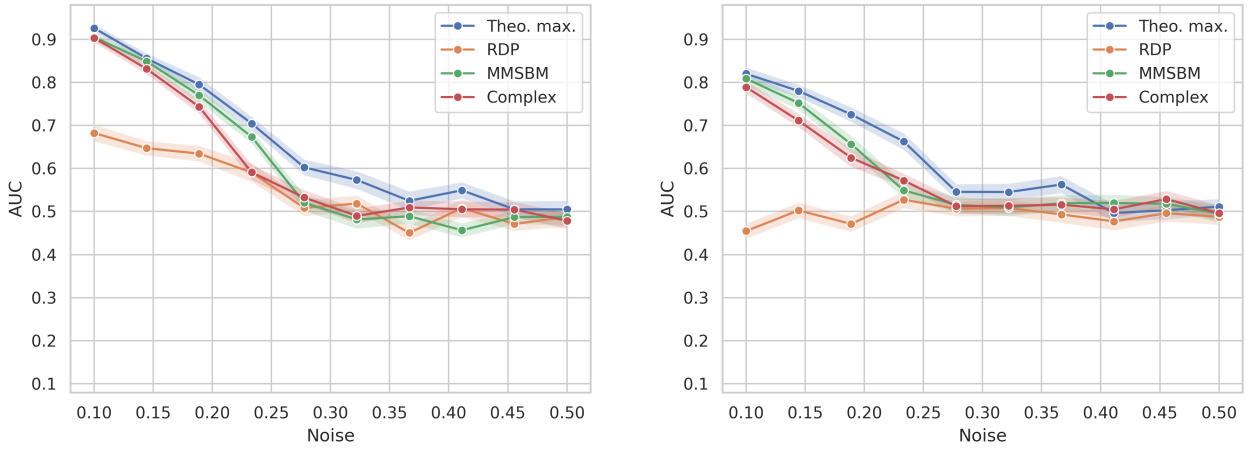
The graph has 3 groups of 20, 40 and 40 nodes with connection probability:

$$P = \begin{bmatrix} 0.4 & 0.9 & 0.0 \\ 0.9 & 0.0 & 0.1 \\ 0.0 & 0.1 & 0.9 \end{bmatrix}$$

The graph has 4 groups of 20, 40, 10 and 30 nodes with connection probability:

$$P = \begin{bmatrix} 0.4 & 0.9 & 0.0 & 0.7 \\ 0.9 & 0.0 & 0.1 & 0.4 \\ 0.0 & 0.1 & 0.9 & 0.1 \\ 0.7 & 0.4 & 0.1 & 0.5 \end{bmatrix}$$

The same trend follows as the RDP performs significantly lower than the other models while MMSBM and the Complex model perform very close to the theoretical maximum.



(a) 3 groups

(b) 4 groups

Figure 4. Comparison in AUC among the different methods for 2 and 3 block networks generated with assortative and disassortative components

3.3.3 Artificial Networks with Degree Correction

The networks analyzed in the previous section are very clearly different from networks we could encounter in real life. In a realistic network we find that even nodes that belong to the same group behave slightly differently. An assumption that we have made is that all nodes in the same group have on average the same number of connections. In real life we would need to account for some differences between nodes. For example, in friendships, some people are more social or popular and might on average have more friends, but that doesn't mean that they belong to a different group. To make the previous graphs closer to real life we incorporate a degree correction. Each node is assigned a factor between $2/3$ and 1, which multiplies the group probabilities of connection. This will make nodes in the same group behave different from each other.

Our 3 models rely on the dimension of their latent space to infer the characteristics of the model. Now that nodes within the same group can behave differently our models might not be able to do the best possible job if only allowed the number of dimensions that have generated the networks. For this we can do a little study plotting the AUC scores for an MMSBM with different numbers of groups or latent dimensions. Increasing the number will typically always increase the

flexibility of the model and therefore we'll expect it to perform better, however our goal is to work with the minimum number of blocks. This will help us understand the structure of our network without overfitting it and keep the computational expense manageable. Fig. 5 shows that there are very small changes in the AUC for different values of K. In the analysis for networks with degree correction we have inferred using one more latent dimension that the networks were originally created with.

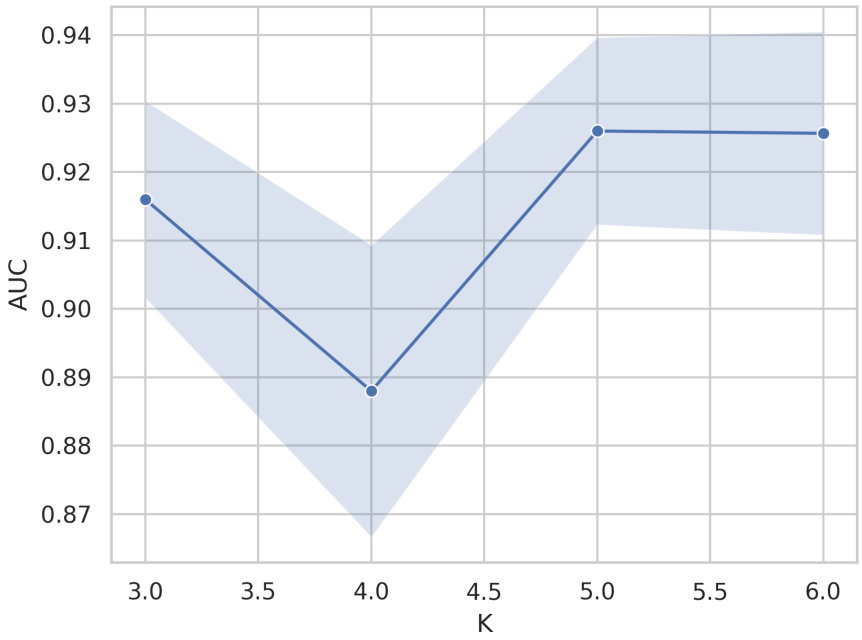


Figure 5. AUC for different values of K in the 3-group artificial network

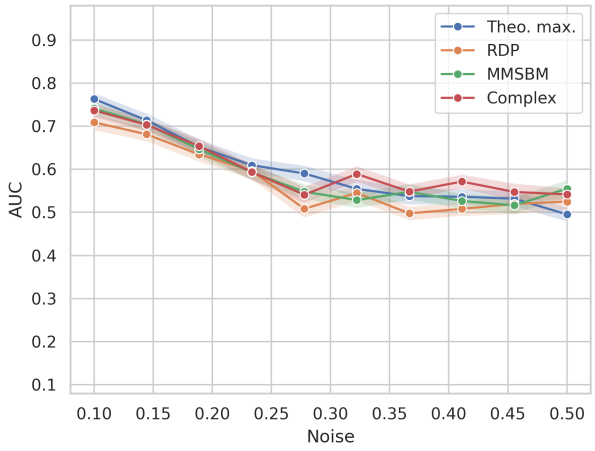
Fig. 6 shows the same behavior as the newtorks in the previous section. For the disassortative we see how the Complex model performs a bit lower than the MMSBM and the theoretical maximum.

When we decrease the connection probability, we see how the models start struggling significantly to grasp the structure of the networks. The difference in performance of the RDP with respect to the others is still noticeable even if not as big as in the earlier plots.

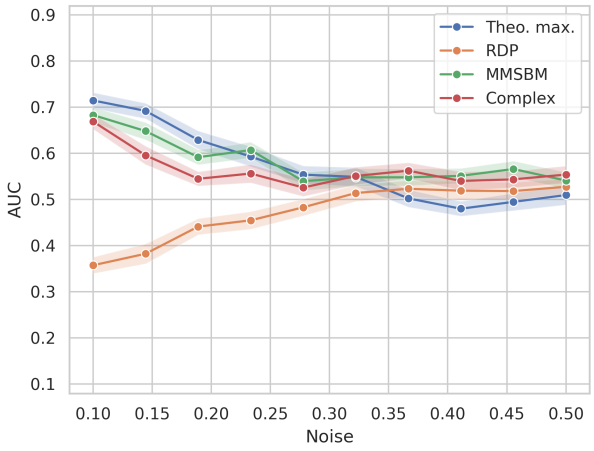
3.3.4 Real-life Networks: Hospital Ward Contacts

The final part in the study of our methods is to apply them to real-life networks.

The first network studied is the "hospital ward dynamic contacts (2010)" [9]. This network consists of 75 nodes representing the people in the hospital ward, 46 of which were health-care workers (HCW) and 29 were patients. Specifically, there are 3 types of HCW represented: paramedical staff, i.e. nurses and nurses' aides, medical doctors, and administrative staff. Although the network is timed, for the purpose of our study it has been simplified so that if contact between two people existed at any time, there will be an edge connecting them without taking into account when they had contact or if they have met multiple times.

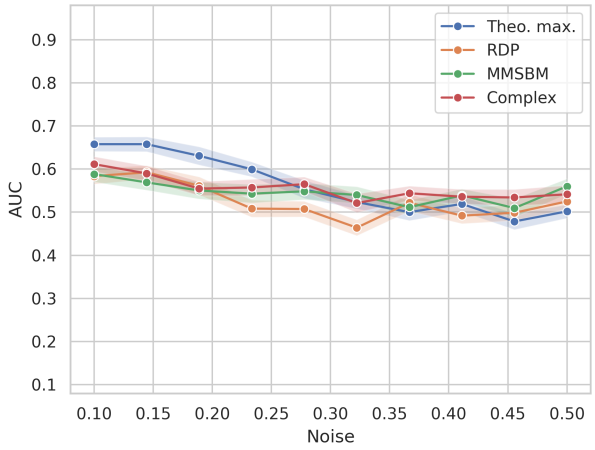


(a) Assortative

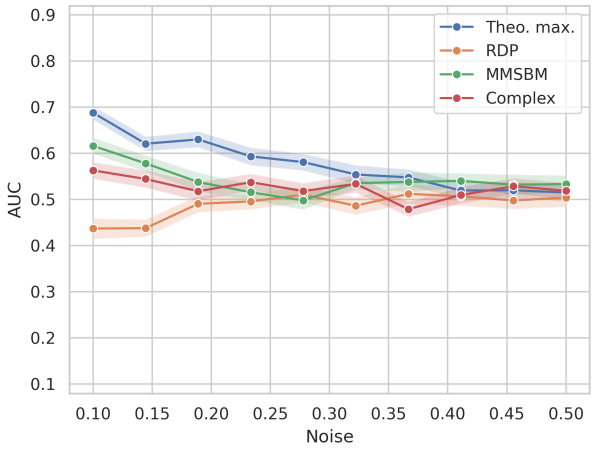


(b) Disassortative

Figure 6. Comparison in AUC among the different methods for 2 block networks generated with probabilities 0.9 and 0.1



(a) Assortative

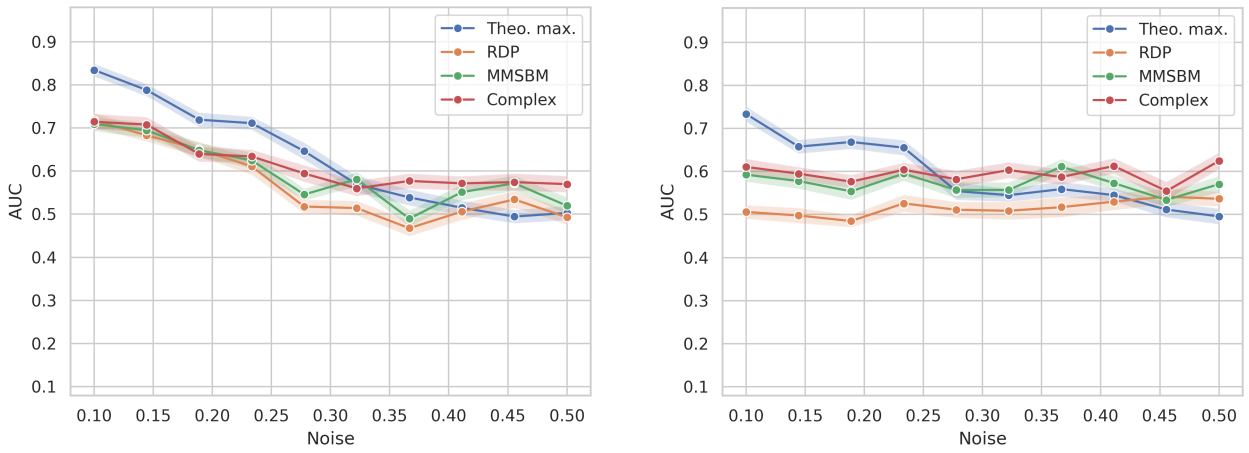


(b) Disassortative

Figure 7. Comparison in AUC among the different methods for 2 block networks generated with probabilities 0.5 and 0.1

Fig. 9 shows the adjacency matrix of the network. At first sight, we can already distinguish a certain block behavior. Knowing the network context, it could be possible that it shows distinctive behavior depending on the role of each person: patients, and the different kinds of wealth-care workers. We hope our models will be able to extract the underlying structure of the network.

We apply RDP, MMSBM and complex inference to the adjacency matrix and obtain the following probability matrices represented in heatmaps in Fig. 10. Since the network already has a natural division into groups we use that as the latent dimensions in all models. We also take the oportunity to not only compare performance between different models but within the same model between different number of latent dimensions. We use K=2 and K=4 since the nodes can already be divided into 2 groups - patients and HCW- and 4 groups- patients, medical aid, medical doctors



(a) 3 groups

(b) 4 groups

Figure 8. Comparison in AUC among the different methods for 3 and 4 block networks generated with assortative and disassortative combination

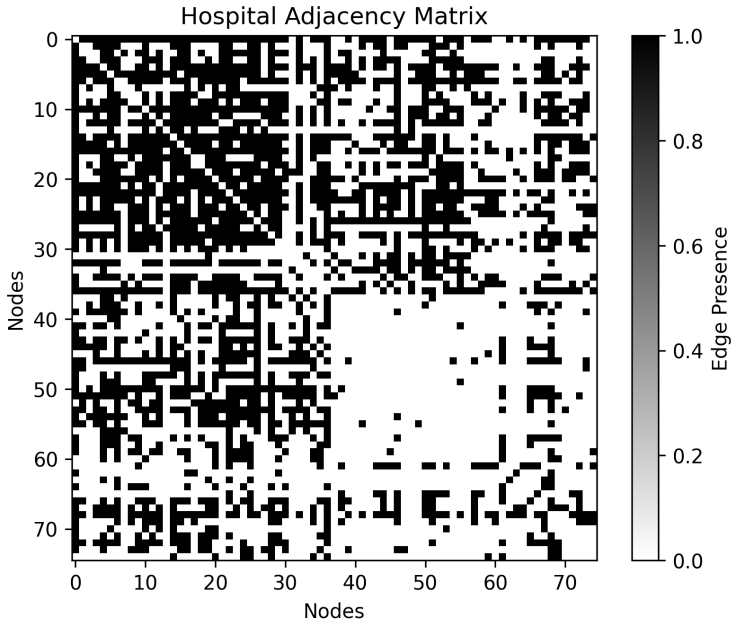
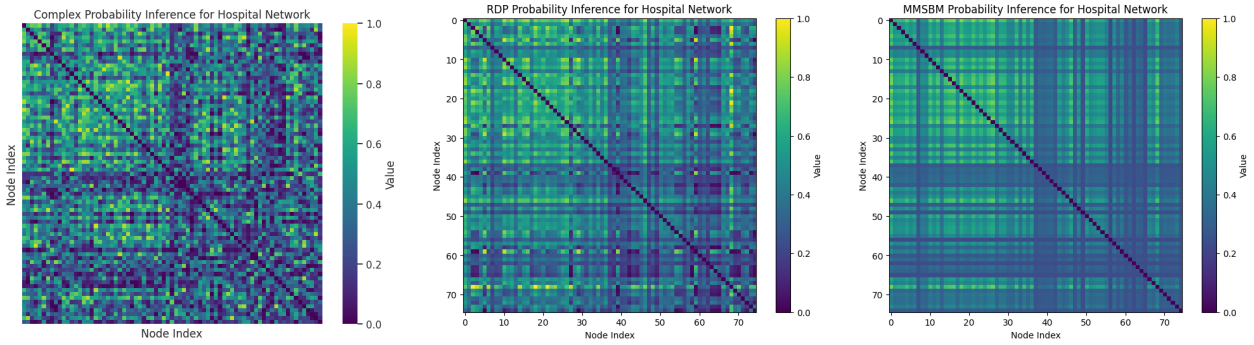


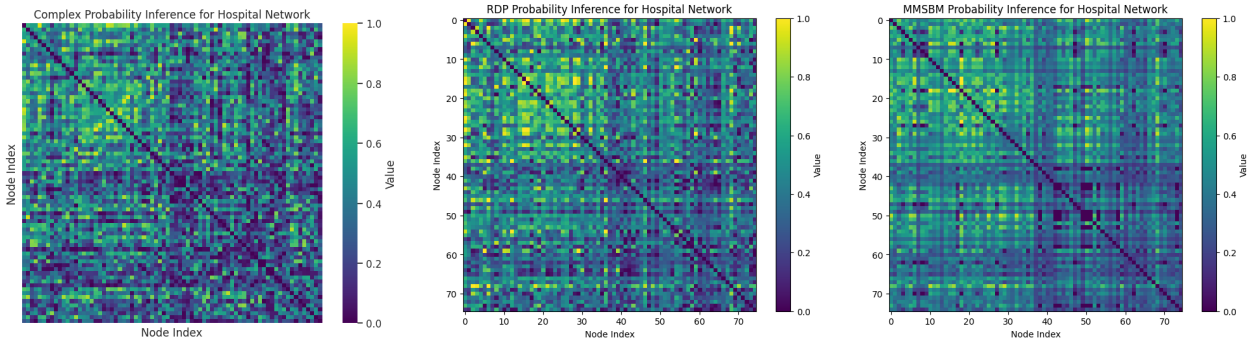
Figure 9. Hospital network adjacency matrix

and administrative staff. From these plots alone we can observe some difference between models. For example we see MMSBM with K=2 has much smoother probabilities more centered around 0.5. This changes when we increase K to 4. Other than this, it is hard to reach any conclusions on the differences between the models and conclude which work best.

Next we focus on understanding how different the predictions each model makes for the probability of each edge are. Fig. 11 shows the scatter plots comparing the probabilities each edge is assigned by each method. On the axis of each plot we see the distribution of the probabilities.



(a) Complex inference (K=2) (b) RDP inference (K=2) (c) SBM inference (K=2)



(d) Complex inference (K=4) (e) RDP inference (K=4) (f) SBM inference (K=4)

Figure 10. Comparison of hospital network edge probability inferences for latent dimensions 2 (top row) and 4 (bottom row).

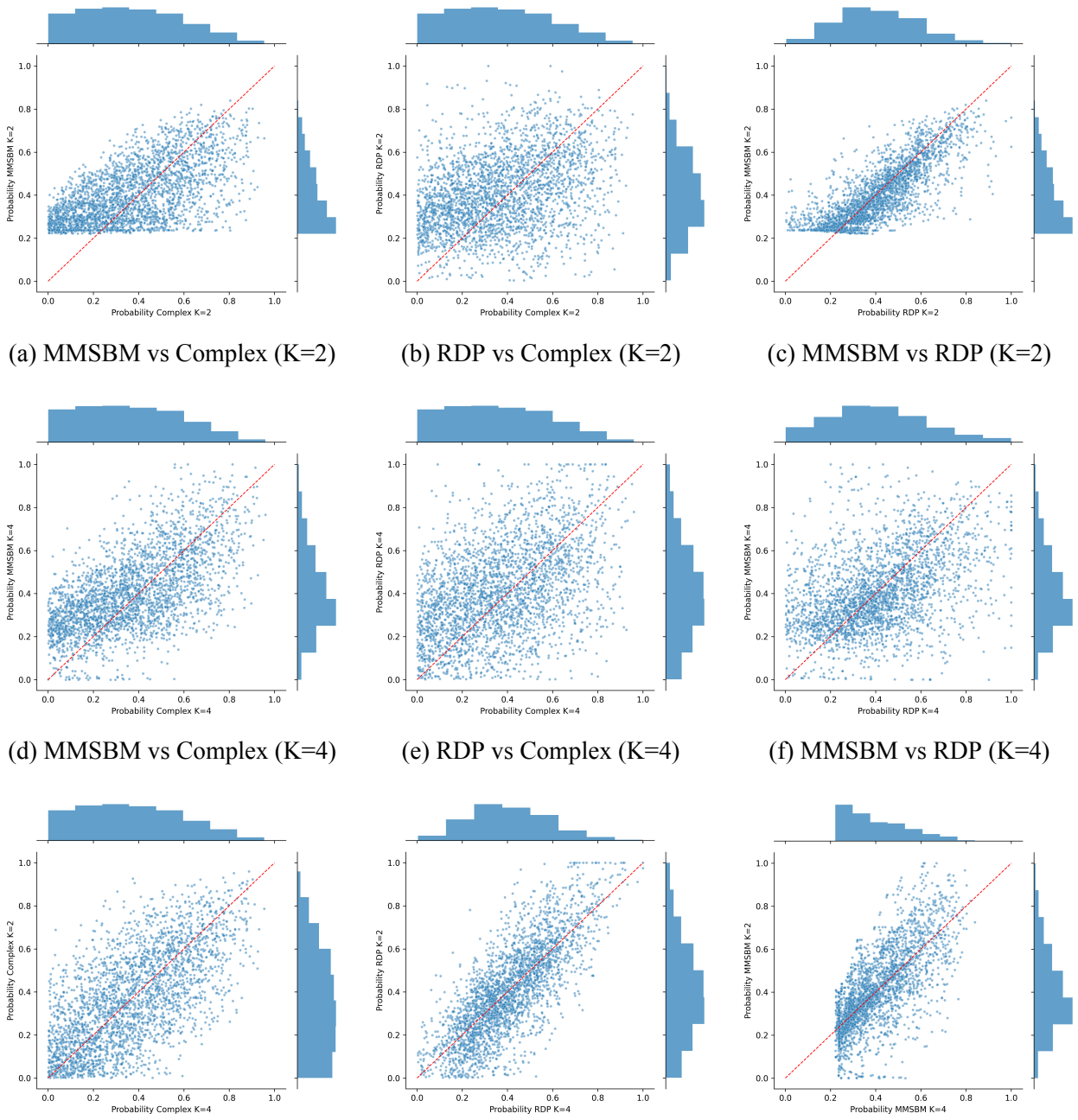
Generally, we see the plots show more or less centered clouds instead of lines or distinct clusters. The distributions of the probabilities are mostly symmetric and unimodal. This means our models are assigning very centered probabilities, around 0.5. Between models we see a slight correlation, meaning most of the edges get either low probabilities in all models or high ones. With respect to the plots comparing each model with different dimensions, we see how there is a stronger correlation than the others, most dots falling close to the $y = x$ line. We also notice how MMSBM with K=2 doesn't assign probabilities lower than 0.2 while for K=4 the range of probabilities goes from 0 to 1.

To be able to truly understand the usefulness of our methods and how they compare to each other it is necessary to quantify their performance by comparing the probabilities to the observed network. For this we use 2 methods.

Log-Likelihood

Our goal is to compare the adjacency matrix A with the inferred probability matrices and quantify how likely it is to get the observed A for each of them. We assume each observation is drawn from a Bernoulli distribution, therefore the probability of observing A_{ij} given P_{ij} is

$$\mathbb{P}(A_{ij} \mid P_{ij}) = P_{ij}^{A_{ij}}(1 - P_{ij})^{1-A_{ij}}$$



(a) MMSBM vs Complex (K=2) (b) RDP vs Complex (K=2) (c) MMSBM vs RDP (K=2)

(d) MMSBM vs Complex (K=4) (e) RDP vs Complex (K=4) (f) MMSBM vs RDP (K=4)

(g) Complex (K=2) vs (K=4) (h) RDP (K=2) vs (K=4) (i) MMSBM (K=2) vs (K=4)

Figure 11. Hospital network edge probability estimates compared among all inference methods and latent dimensions with the marginal probability distributions

And the likelihood of the entire dataset is:

$$\mathcal{L}(P;A) = \prod_{i=1}^{N}\prod_{j=1}^{N} P_{ij}^{A_{ij}}(1-P_{ij})^{1-A_{ij}}$$

We get to the log-likelihood by taking the logarithm.

$$\log \mathcal{L}(P; A) = \sum_{i=1}^{N} \sum_{j=1}^{N} [A_{ij} \log(P_{ij}) + (1 - A_{ij}) \log(1 - P_{ij})]$$

A good P will have P_{ij} close to 1 when $A_{ij} = 1$ and close to 0 when $A_{ij} = 0$. Therefore, we want the values in the logarithms to be close to 1. The higher or less negative the log-likelihood value, the better the model performs.

A completely non-informative model, with all probabilities = 0.5, will give us a reference point to compare the log-likelihoods for our model.

We calculate for P such that all $P_{ij} = 0.5$

$$\log \mathcal{L}(P; A) = \sum_{i=1}^{N} \sum_{j=1}^{N} [A_{ij} \log(P_{ij}) + (1 - A_{ij}) \log(1 - P_{ij})]$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} [A_{ij} \log(0.5) + (1 - A_{ij}) \log(0.5)] = -N^2 \log(2) = -3898.95289$$

The calculation of the logarithmic likelihoods of our models in Table 1 shows the only model that performs worse than our reference is the RDP. The complex inference is the best-performing model. It should be taken into account how the number of latent dimensions affects the results measured as log-likelihood. We expect models with more dimensions to perform better as they are more flexible and can capture more complex relationships. It should be noted that although we set the vector dimensions in all models to 2 and 4, each models particularities make the real number of dimensions different. The normalisation in MMSBM makes the true dimension $k - 1$ while the complex model's true dimension is $2k$. Comparing these true dimensions brings to light different things. The small difference in log-likelihood between the 2 dimensions for the complex model makes us think that the extra flexibility from the two extra dimensions isn't necessary. We see how the RDP, even with 4 dimensions performs worse than the equivalent complex model. This could be tied to the findings of the previous sections and how RDP fails to model disassortative behaviour. Also worth noting is that the MMSBM gets very reasonable results even with less dimensions than the RDP.

Table 1. Log-Likelihoods (Best to Worst)

Method	K	Real Dimensions	Log-Likelihood
Complex	2	4	-2795.182612
Complex	4	8	-2800.731994
MMSBM	4	3	-3314.590542
MMSBM	2	1	-3518.140931
RDP	4	4	-3967.870212
RDP	2	2	-4013.696038

AUC

Finally, to show how the models actually predict missing links, we calculate the AUC in the same way as we have been analyzing the artificial networks. We introduce noise to our network and calculate AUC for each value. Since we only have one network, in order to give an average and standard deviation, we create different noisy networks for every level of noise.

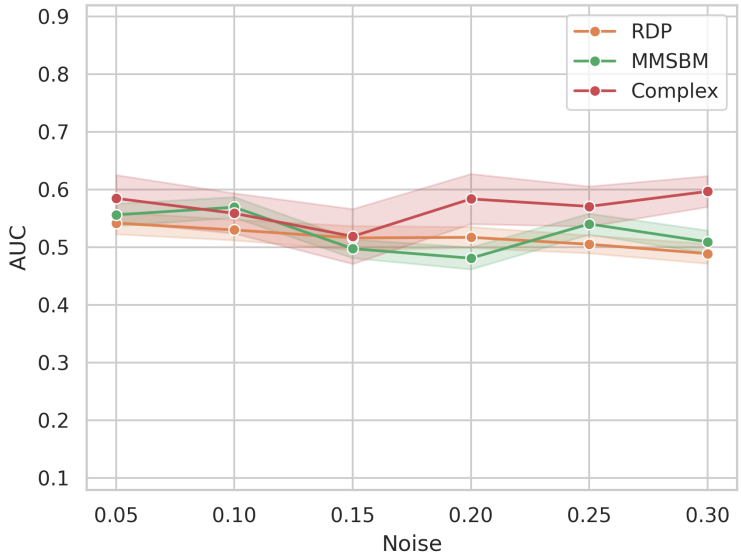


Figure 12. Hospital network AUC comparison between models

Fig. 12 shows the AUC plot. All the models show the same results very close to 0.5 for all levels of noise. This implies that they didn't infer any particular structure in the network. This ties back to what we observed in the scatter plots: most of the infered probabilities are very centered around 0.5. This is not informative enough. When calculating AUC we need the probabilities to be more extreme so that the ordering is not left to chance. It is shown that none of our models is able to provide good predictions.

3.3.5 Real-life Networks: Kenyan Household Network

The second network here shows the contacts measured between members of 5 households of rural Kenya [10]. From the visualization of the adjacency matrix in Fig. 13 we discover a mostly assortative behaviour. We analyse it in the same way as the hospital network. The inference study has also been done with K=2 and K=4.

The heatmaps in Fig. 14 show once again how MMSBM with K=2 shows very centered probabilities while other models show more extreme probabilities. At first glance, it looks like the block structure has been captured by all models.

Fig. 15 shows the scattered inferred probabilities. This time the only model showing a unimodal centered distribution is MMSBM with K=2. All other models now have a more bimodal distribution of probabilities, having the biggest amount of predictions in the extremes: close to 0

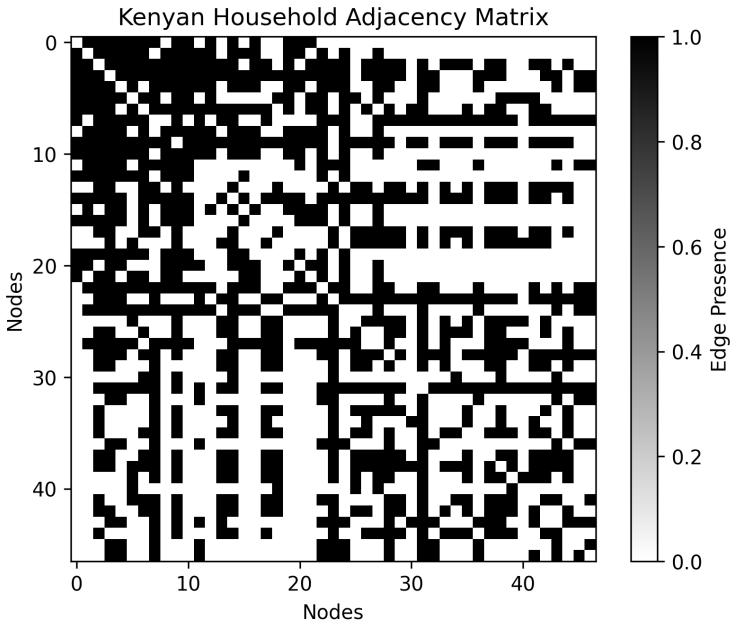
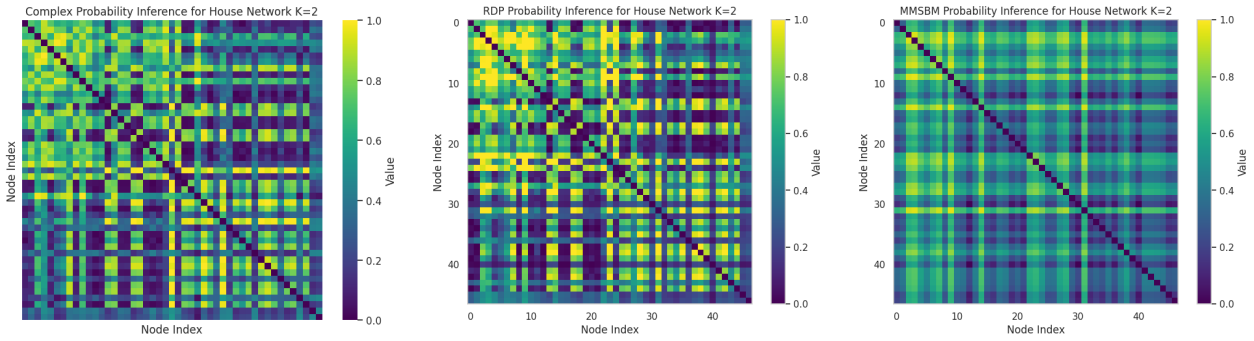
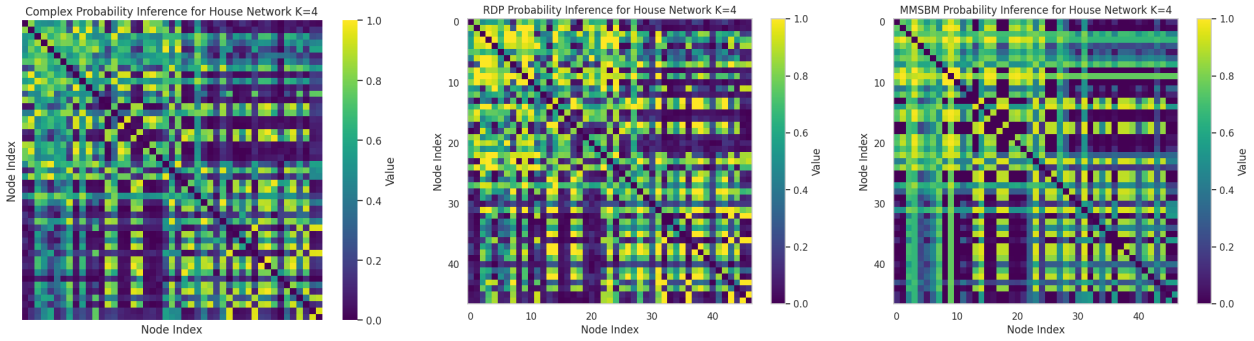


Figure 13. Kenyan Household Network adjacency matrix



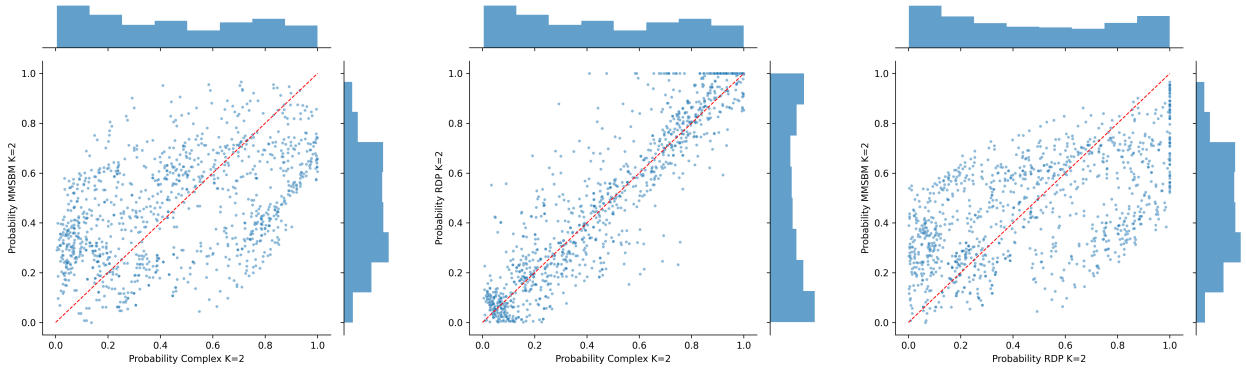
(a) Complex inference (K=2) (b) RDP inference (K=2) (c) SBM inference (K=2)



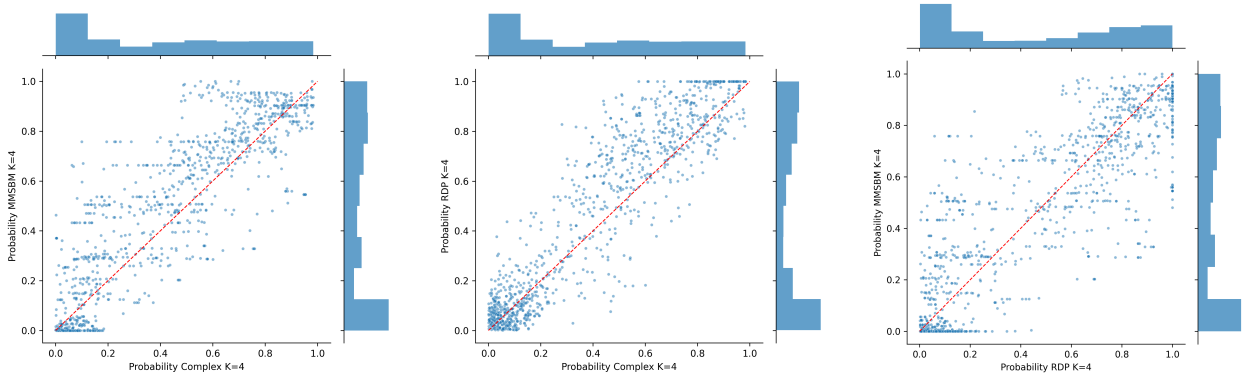
(d) Complex inference (K=4) (e) RDP inference (K=4) (f) SBM inference (K=4)

Figure 14. Comparison of house network edge probability inferences for latent dimensions 2 (top row) and 4 (bottom row).

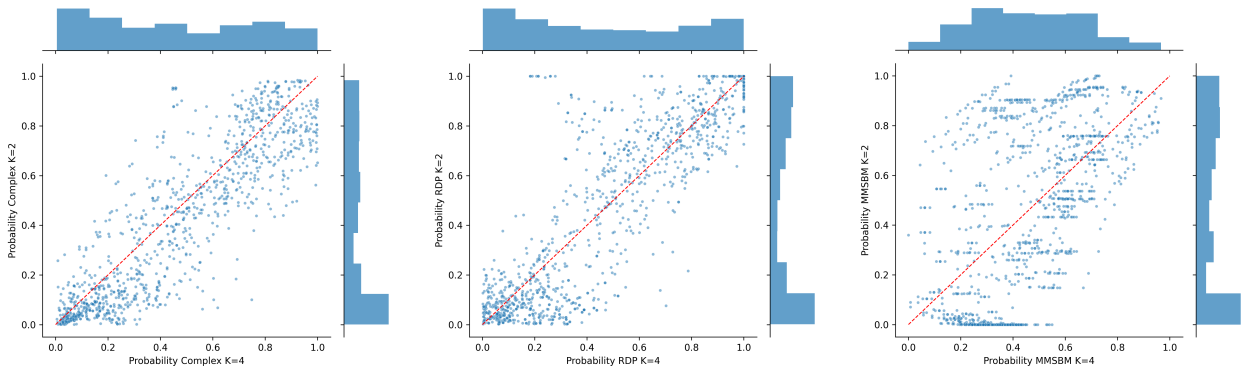
and close to 1. We also observe the scatter plots between RDP and Complex model show most dots close to the $y = x$ meaning that they do similar predictions for each edge.



(a) MMSBM vs Complex (K=2) (b) RDP vs Complex (K=2) (c) MMSBM vs RDP (K=2)



(d) MMSBM vs Complex (K=4) (e) RDP vs Complex (K=4) (f) MMSBM vs RDP (K=4)



(g) Complex (K=2) vs (K=4) (h) RDP (K=2) vs (K=4) (i) MMSBM (K=2) vs (K=4)

Figure 15. Household network edge probability estimates compared among all inference methods and latent dimensions with the marginal probability distributions

Log-Likelihood

This time our value of reference for the log-likelihood is calculated for $n = 47$ as:

$$\log \mathcal{L}(P; A) = -N^2 \log(2) = -47^2 \log(2) = -1531.162122$$

In Table 2 we compare all methods. This time it becomes obvious that the models with K=4 perform better than for K=2. The complex model, having more parameters, has a clear advantage over the others. However, also the MMSBM with K=4 gets remarkable results, with less free parameters than the other models. No model performs worse than the reference value but MMSBM with K=2 performs the worst with a clear difference with respect to the others.

Table 2. Log-Likelihoods for the household network (Best to Worst)

Method	K	Real Dimensions	Log-Likelihood
Complex	4	8	-708.066610
MMSBM	4	3	-794.222314
RDP	4	4	-817.214034
Complex	2	4	-860.322322
RDP	2	2	-862.569788
MMSBM	2	1	-1278.957373

AUC

Fig. 16 shows how all models have very similar results although this time they get very good results for the AUC values with little noise and decrease as expected for lower values. This matches with what we expected: now that our models predict more extreme probabilities, in the AUC calculations they can produce a good ordering of the edges. We cannot get any conclusions on what method works best.

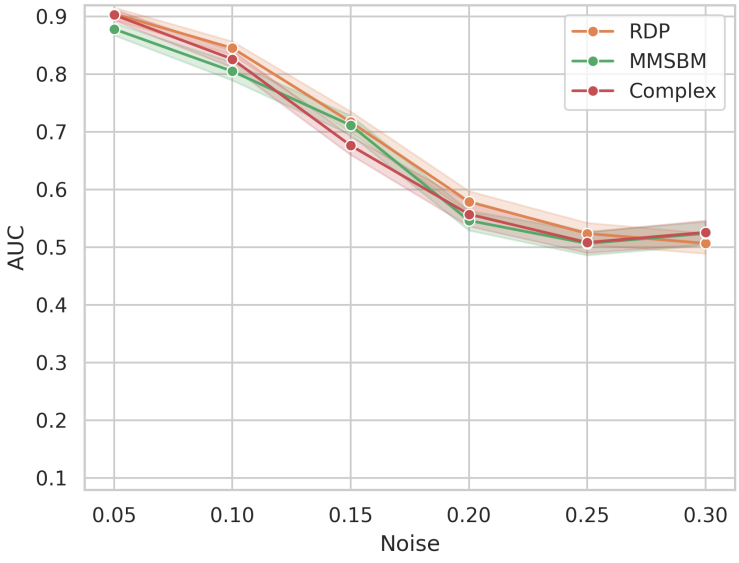


Figure 16. Hospital network AUC comparison between models

4 Conclusion

The primary goal of this study was to compare the effectiveness of the MMSBM and the RDP frameworks, and to prove that the latter fails to model disassortativity. Through theoretical analysis and practical experiments, we have shown that while MMSBM can successfully represent both assortative and disassortative structures, the standard RDP struggles with disassortativity due to fundamental limitations in its design. In particular, the way it calculates inner products limits the kinds of connection patterns it can represent. We showed clear evidence contradicting the claim by Athreya, Fishkind, Tang, *et al.* [6] that RDP is a generalization of MMSBM, since MMSBM is able to model any kind of network regardless of whether it is assortative or disassortative.

To solve this problem, we introduced a new version of the RDP that uses complex numbers for the node vectors. This change allows for negative inner products, providing more flexibility in how connections are modeled. Thanks to this, the model can now represent disassortative patterns. By using complex-valued embeddings, our method keeps the simplicity of the original RDP, while making it suitable for a wider range of network structures.

We've tried to develop equations for the parameters in the complex model using the expectation-maximization algorithm, however, the implementation did not succeed. Although we don't know the exact reason, it is most likely due to a broken assumption in the math behind the model. Solving this issue is an important step for future work, as a working version would allow for efficient and interpretable inference on real-world data. The solution to this limitation is left for future research.

In our experiments, we compared the performance of all three models on synthetic and real-world networks. RDP failed to model even the simplest disassortative networks, while MMSBM and the Complex RDP both performed well for assortative and disassortative structures. When tested on more complex networks with a mix of assortative and disassortative blocks, the RDP showed lower accuracy compared to the other two models, due to the disassortative patterns it could not capture. This shows that we need models that can handle both types of connection patterns to make reliable inferences.

In the real-world networks we studied, we also saw that sometimes the inference failed to give meaningful predictions. This points to the importance of choosing the right model based on the network's structure. However, for an assortative network, all three methods gave useful results.

In conclusion, our work highlights the limits of the standard RDP and the need for more expressive, yet simple, models to understand network structures. Our complex-valued version of RDP shows potential as a practical solution, especially in cases where disassortative relationships are important. Even though some implementation challenges remain, this approach adds to the growing effort to build flexible and efficient tools for graph inference. With further development, we believe our method could become a valuable tool for analyzing networks in many different fields.

5 Bibliography

[1] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Engineering Bulletin*, vol. 40, no. 3, pp. 52–74, 2017.

[2] S. J. Young and E. R. Scheinerman, "Random dot product graph models for social networks," in *Algorithms and Models for the Web-Graph (WAW 2007)*, ser. Lecture Notes in Computer Science, vol. 4863, Springer, 2007, pp. 138–149. DOI: 10.1007/978-3-540-77004-6_11.

[3] M. E. J. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, p. 026 126, 2003. DOI: 10.1103/PhysRevE.67.026126.

[4] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research*, vol. 9, T. Jaakkola, Ed., pp. 1981–2014, 2008. [Online]. Available: http://www.jmlr.org/papers/volume9/airoldi08a/airoldi08a.pdf.

[5] A. Godoy-Lorite *et al.*, "Accurate and scalable social recommendation using mixed-membership stochastic block models," *Proceedings of the National Academy of Sciences*, vol. 113, no. 50, pp. 14 207–14 212, 2016. DOI: 10.1073/pnas.1606316113.

[6] A. Athreya, D. E. Fishkind, M. Tang, *et al.*, "Statistical inference on random dot product graphs: A survey," *Journal of Machine Learning Research*, vol. 18, 226:1–226:92, 2018. [Online]. Available: https://jmlr.org/papers/volume18/17-448/17-448.pdf.

[7] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983. DOI: 10.1016/0378-8733(83)90021-7.

[8] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. DOI: 10.1016/j.patrec.2005.10.010.

[9] P. Vanhems, A. Barrat, C. Cattuto, *et al.*, "Estimating potential infection transmission routes in hospital wards using wearable proximity sensors," *PLoS ONE*, vol. 8, no. 9, e73970, 2013. DOI: 10.1371/journal.pone.0073970.

[10] M. C. Kiti, T. M. Kinyanjui, D. C. Koech, P. K. Munywoki, G. F. Medley, and D. J. Nokes, "Quantifying social contacts in a household setting of rural kenya using wearable proximity sensors," *EPJ Data Science*, vol. 5, no. 1, p. 21, 2016. DOI: 10.1140/epjds/s13688-016-0084-2. [Online]. Available: https://doi.org/10.1140/epjds/s13688-016-0084-2.

Appendices

A Equations for the Complex model

To implement this method we attempt to develop equations for the module and phase of each component in the vectors.

$\vec{z_i}$ is the vector that represents node i. $z_{i\alpha} \in \mathbb{C}\ \alpha = 1, ..., k$

$$\vec{z_i} \cdot \vec{z_i^*} = 1 \qquad \sum_\alpha z_{i\alpha} z_{i\alpha}^* = 1$$

$$p(A_{ij} = 1 | \vec{z_i} \vec{z_j}) = (\vec{z_i} \cdot \vec{z_j}) (\vec{z_i} \cdot \vec{z_j})^* = \sum_{\alpha\beta} z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^*$$

$$p(A_{ij} = 0 | \vec{z_i} \vec{z_j}) = 1 - p(A_{ij} = 1 | \vec{z_i} \vec{z_j}) = 1 - \sum_{\alpha\beta} z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^* = \sum_{\alpha\beta} \frac{1}{K^2} - z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^*$$

Likelihood:

$$L = p(A | \{\vec{z_i}\}) = \prod_{i<j} p(A_{ij} | \vec{z_i} \vec{z_j}) = \prod_{i<j} p_{ij}^{A_{ij}} (1 - p_{ij})^{(1 - A_{ij})}$$

Log-likelihood:

$$\begin{aligned} \log L &= \sum_{i<j} A_{ij} \log p_{ij} + (1 - A_{ij}) \log (1 - p_{ij}) \\ &= \sum_{i<j} A_{ij} \log \left(\sum_{\alpha\beta} z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^* \right) + (1 - A_{ij}) \log \left(\sum_{\alpha\beta} \frac{1}{K^2} - z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^* \right) \end{aligned}$$

We introduce auxiliary distributions $\omega_{\alpha\beta}(ij)$ and $\tilde{\omega}_{\alpha\beta}(ij)$ such that $\sum_{\alpha\beta} \omega_{\alpha\beta}(ij) = 1\ \forall ij$ and $\sum_{\alpha\beta} \tilde{\omega}_{\alpha\beta}(ij) = 1\ \forall ij$. By Jensen's inequality:

$$\log \left(\sum_{\alpha\beta} \omega_{\alpha\beta}(ij) \frac{z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^*}{\omega_{\alpha\beta}(ij)} \right) \geq \sum_{\alpha\beta} \omega_{\alpha\beta}(ij) \log \left(\frac{z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^*}{\omega_{\alpha\beta}(ij)} \right)$$

Where $\omega_{\alpha\beta}(ij)$ is a complex number but $\frac{z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^*}{\omega_{\alpha\beta}(ij)}$ is a real positive number. The inequality holds; later will prove that it's true for the equal sign.

$$\log \left(\sum_{\alpha\beta} \frac{\frac{1}{K^2} - z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^*}{\tilde{\omega}_{\alpha\beta}(ij)} \right) \geq \sum_{\alpha\beta} \tilde{\omega}_{\alpha\beta}(ij) \log \left(\frac{\frac{1}{K^2} - z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^*}{\tilde{\omega}_{\alpha\beta}(ij)} \right)$$

Now we use the exponential form for the complex number $z_{i\alpha} = r_{i\alpha} e^{i\phi_{i\alpha}}$ so that $z_{i\alpha} z_{j\alpha} z_{i\beta}^* z_{j\beta}^* = r_{i\alpha} r_{j\alpha} r_{i\beta} r_{j\beta} e^{i(\phi_{i\alpha} + \phi_{j\alpha} - \phi_{i\beta} - \phi_{i\beta})}$.

We define $R_{ij\alpha\beta} \equiv r_{i\alpha} r_{j\alpha} r_{i\beta} r_{j\beta}$ which is a real number. It's symmetric under permutation of indices ij and $\alpha\beta$. $R_{ij\alpha\beta} = R_{ji\alpha\beta} = R_{ij\beta\alpha} = R_{ji\beta\alpha}$.

We also define $\Delta\Phi_{ij\alpha\beta} = \phi_{i\alpha} + \phi_{j\alpha} - \phi_{i\beta} - \phi_{i\beta}$. It's symmetric on the permutation of ij: $\Delta\Phi_{ij\alpha\beta} = \Delta\Phi_{ji\alpha\beta}$ and antisymmetric for the permutation of $\alpha\beta$: $\Delta\Phi_{ij\alpha\beta} = -\Delta\Phi_{ij\beta\alpha}$

We then define:

$$\mathcal{L} = \sum_{i<j} A_{ij} \sum_{\alpha\beta} \omega_{\alpha\beta}(ij) \log\left(\frac{R_{ij\alpha\beta}e^{i\Delta\Phi_{ij\alpha\beta}}}{\omega_{\alpha\beta}(ij)}\right) + (1 - A_{ij}) \sum_{\alpha\beta} \tilde{\omega}_{\alpha\beta}(ij) \log\left(\frac{\frac{1}{K^2} - R_{ij\alpha\beta}e^{i\Delta\Phi_{ij\alpha\beta}}}{\tilde{\omega}_{\alpha\beta}(ij)}\right)$$

We want to look for maxima of the $\mathcal{L}$ function so that $\log L \geq \mathcal{L}$

$$\frac{\partial\mathcal{L}}{\partial\omega_{\gamma\delta}(km)} = A_{km} \log\frac{R_{km\gamma\delta}e^{i\Delta\Phi_{km\gamma\delta}}}{\omega_{\gamma\delta}(km)} - A_{km} = \lambda_{\gamma\delta km}$$

Applying normalization constraint $\sum_{\gamma\delta}\omega_{\gamma\delta}(km) = 1$ we finally get:

$$\omega_{\gamma\delta}(km) = \frac{R_{km\gamma\delta}e^{i\Delta\Phi_{km\gamma\delta}}}{\sum_{\gamma'\delta'} R_{km\gamma'\delta'}e^{i\Delta\Phi_{km\gamma'\delta'}}} = \frac{R_{km\gamma\delta}e^{i\Delta\Phi_{km\gamma\delta}}}{p_{ij}}$$

With equivalent calculations, we get:

$$\tilde{\omega}_{\gamma\delta}(km) = \frac{\frac{1}{K^2} - R_{km\gamma\delta}e^{i\Delta\Phi_{km\gamma\delta}}}{1 - p_{ij}}$$

Now we compute the derivatives according to the normalization constraints.

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\phi_{k\gamma}} = \sum_{j\neq k}\Bigg[A_{kj}\sum_{\alpha}\left(-\omega_{\alpha\gamma}(kj)\,i + \omega_{\gamma\alpha}(kj)\,i\right)\\ + (1 - A_{kj})\sum_{\alpha}\left(\tilde{\omega}_{\alpha\gamma}(kj)\frac{iR_{kj\alpha\gamma}e^{i\Delta\Phi_{kj\alpha\gamma}}}{\frac{1}{K^2} - R_{kj\alpha\gamma}e^{i\Delta\Phi_{kj\alpha\gamma}}} + \tilde{\omega}_{\gamma\alpha}(kj)\frac{-iR_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}}{\frac{1}{K^2} - R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}}\right)\Bigg] = 0\end{aligned}$$

We incorporate the equivalences for $\omega_{\alpha\beta}(ij)$ and $\tilde{\omega}_{\alpha\beta}(ij)$.

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\phi_{k\gamma}} &= \sum_{j\neq k}\Bigg[A_{kj}\sum_{\alpha}\left(-i\frac{R_{kj\alpha\gamma}e^{i\Delta\Phi_{kj\alpha\gamma}}}{p_{kj}} + i\frac{R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}}{p_{kj}}\right)\\ &\quad + (1 - A_{kj})\sum_{\alpha}\left(-i\frac{R_{kj\alpha\gamma}e^{i\Delta\Phi_{kj\alpha\gamma}}}{1 - p_{kj}} + i\frac{R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}}{1 - p_{kj}}\right)\Bigg]\\ &= \sum_{j\neq k}A_{kj}\sum_{\alpha}\frac{iR_{kj\alpha\gamma}}{p_{kj}}\left(-e^{-i\Delta\Phi_{kj\gamma\alpha}} + e^{i\Delta\Phi_{kj\gamma\alpha}}\right)\\ &\quad + (1 - A_{kj})\sum_{\alpha}\frac{iR_{kj\alpha\gamma}}{1 - p_{kj}}\left(e^{-i\Delta\Phi_{kj\gamma\alpha}} - e^{i\Delta\Phi_{kj\gamma\alpha}}\right) = 0\end{aligned}$$

We know $\sin x = \frac{e^{ix}-e^{-ix}}{2i}$ therefore $e^{i\Delta\Phi_{kj\gamma\alpha}} - e^{-i\Delta\Phi_{kj\gamma\alpha}} = 2i\sin\Delta\Phi_{kj\gamma\alpha}$

$$\frac{\partial\mathcal{L}}{\partial\phi_{k\gamma}} = \sum_{j\neq k} A_{kj} \sum_{\alpha} \frac{R_{kj\alpha\gamma}}{p_{kj}}(-2\sin\Delta\Phi_{kj\gamma\alpha}) - (1-A_{kj})\sum_{\alpha}\frac{R_{kj\alpha\gamma}}{1-p_{kj}}(-2\sin\Delta\Phi_{kj\gamma\alpha}) = 0$$

In order to get an expression for $\phi_{k\gamma}$ we use:

$$\sin\Delta\Phi_{kj\gamma\alpha} = \sin\left(\phi_{k\gamma} + \phi_{j\gamma} - \phi_{k\alpha} - \phi_{j\alpha}\right) = \sin\phi_{k\gamma}\cos\Delta\Phi'_{kj\gamma\alpha} + \cos\phi_{k\gamma}\sin\Delta\Phi'_{kj\gamma\alpha}$$

where $\Delta\Phi'_{kj\gamma\alpha} = \phi_{j\gamma} - \phi_{k\alpha} - \phi_{j\alpha}$ grouping terms we get:

$$\begin{aligned}
-\frac{\partial\mathcal{L}}{\partial\phi_{k\gamma}} &= \sum_{j\neq k} A_{kj}\sum_{\alpha} 2\frac{R_{kj\alpha\gamma}}{p_{kj}}(\sin\phi_{k\gamma}\cos\Delta\Phi'_{kj\gamma\alpha} + \cos\phi_{k\gamma}\sin\Delta\Phi'_{kj\gamma\alpha}) \\
&\quad - (1-A_{kj})\sum_{\alpha} 2\frac{R_{kj\alpha\gamma}}{1-p_{kj}}(\sin\phi_{k\gamma}\cos\Delta\Phi'_{kj\gamma\alpha} + \cos\phi_{k\gamma}\sin\Delta\Phi'_{kj\gamma\alpha}) \\
&= \sin\phi_{k\gamma}\left[\sum_{j\neq k}\frac{A_{kj}}{p_{kj}\sum_{\alpha}R_{kj\alpha\gamma}\cos\Delta\Phi'_{kj\gamma\alpha}} - \frac{1-A_{kj}}{1-p_{kj}}\sum_{\alpha}R_{kj\alpha\gamma}\cos\Delta\Phi'_{kj\gamma\alpha}\right] = 0
\end{aligned}$$

We finally obtain:

$$\tan\phi_{k\gamma} = -\frac{\sum_{j\neq k}\left(\frac{A_{kj}}{p_{kj}} - \frac{1-A_{kj}}{1-p_{kj}}\sum_{\alpha}R_{kj\gamma\alpha}\sin\Delta\Phi'_{kj\gamma\alpha}\right)}{\sum_{j\neq k}\left(\frac{A_{kj}}{p_{kj}} - \frac{1-A_{kj}}{1-p_{kj}}\sum_{\alpha}R_{kj\gamma\alpha}\cos\Delta\Phi'_{kj\gamma\alpha}\right)}$$

Equation for parameter $r_{k\gamma}$

$$\frac{\partial\mathcal{L}}{\partial r_{k\gamma}} = 2\lambda_{k\gamma}r_{k\gamma}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial r_{k\gamma}} &= \sum_{j\neq k} A_{kj}\sum_{\alpha}\left(\frac{\omega_{\gamma\alpha}(kj)}{r_{k\gamma}} + \frac{\omega_{\alpha\gamma}(kj)}{r_{k\gamma}}\right) \\
&\quad + \sum_{j\neq k}(1-A_{kj})\sum_{\alpha}\left(\frac{\tilde{\omega}_{\gamma\alpha}(kj)}{r_{k\gamma}}\cdot\frac{-R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}}{\frac{1}{K^2} - R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}} + \frac{\tilde{\omega}_{\alpha\gamma}(kj)}{r_{k\gamma}}\cdot\frac{-R_{kj\alpha\gamma}e^{i\Delta\Phi_{kj\alpha\gamma}}}{\frac{1}{K^2} - R_{kj\alpha\gamma}e^{i\Delta\Phi_{kj\alpha\gamma}}}\right)
\end{aligned}$$

Using the definition for $\omega_{\alpha\gamma}(kj)$ and $\tilde{\omega}_{\alpha\gamma}(kj)$:

$$\frac{\tilde{\omega}_{\alpha\gamma}(kj)}{r_{r_{k\gamma}}}\cdot\frac{-R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}}{\frac{1}{K^2} - R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}} = \frac{1}{r_{k\gamma}}\cdot\frac{-R_{kj\gamma\alpha}e^{i\Delta\Phi_{kj\gamma\alpha}}}{1-p_{kj}} = \frac{1}{r_{k\gamma}}\cdot\frac{p_{kj}}{1-p_{kj}}\cdot\omega_{\gamma\alpha}(kj)$$

$$\frac{\partial \mathcal{L}}{\partial r_{k\gamma}} = \sum_{j\neq k} \frac{A_{kj}}{r_{k\gamma}} \sum_{\alpha} (\omega_{\gamma\alpha}(kj) + \omega_{\alpha\gamma}(kj)) - \sum_{j\neq k} \frac{(1-A_{kj})}{r_{k\gamma}} \frac{p_{kj}}{(1-p_{kj})} \sum_{\alpha} (\omega_{\gamma\alpha}(kj) + \omega_{\alpha\gamma}(kj))$$

$$2\lambda_{kj} r_{kj}^2 = \sum_{j\neq k} A_{kj} \sum_{\alpha} (\omega_{\gamma\alpha}(kj) + \omega_{\alpha\gamma}(kj)) - \sum_{j\neq k} (1-A_{kj}) \frac{p_{kj}}{(1-p_{kj})} \sum_{\alpha} (\omega_{\gamma\alpha}(kj) + \omega_{\alpha\gamma}(kj))$$

Since $\sum_{\alpha} r_{i\alpha}^2 = 1$ we have that

$$\sum_{\alpha\gamma} (\omega_{\gamma\alpha}(kj) + \omega_{\alpha\gamma}(kj)) = \sum_{\alpha\gamma} \frac{R_{kj\gamma\alpha} e^{\Delta\Phi_{kj\gamma\alpha}} + R_{kj\alpha\gamma} e^{\Delta\Phi_{kj\alpha\gamma}}}{\sum_{\alpha'\gamma'} R_{kj\alpha'\gamma'} e^{\Delta\Phi_{kj\alpha'\gamma'}}} = 2$$

$$2\lambda_{kj} = \sum_{j\neq k} [A_{kj} - (1-A_{kj}) \frac{p_{kj}}{1-p_{kj}}] \sum_{\alpha\gamma} (\omega_{\gamma\alpha}(kj) + \omega_{\alpha\gamma}(kj))$$

$$\lambda_{kj} = \sum_{j\neq k} [A_{kj} - (1-A_{kj}) \frac{p_{kj}}{1-p_{kj}}]$$

Therefore:

$$r_{kj}^2 = \frac{\sum_{j\neq k} [(A_{kj} - (1-A_{kj}) \frac{p_{kj}}{1-p_{kj}}) \sum_{\alpha} Re(\omega_{\alpha\gamma}(kj))]}{\sum_{j\neq k} [(A_{kj} - (1-A_{kj}) \frac{p_{kj}}{1-p_{kj}})}$$

B Programming code

All methods have been implemented in Python using Jupiter notebooks. Several notebooks were used to create the networks and help analyze the results. The codes for the three models are available in the following GitHub repository: https://github.com/illoberaq/tfg_illoberaq.git.