Università degli Studi di Padova

Department of Mathematics 'Tullio Levi-Civita'

Master of Science in Data Science

A New Approach for Community Detection in Multilayer Networks

Supervisor: Prof. Francesco Rinaldi

Department of Mathematics

Co-Supervisor: Caterina De Bacco

Max Planck Institute for Intelligent Systems

Student: Martina Contisciani

N. 1179040

Academic Year 2018/2019

Alla mia famiglia

Contents

Introduction

Network science has been described by the United States National Research Council as 'the study of network representations of physical, biological, and social phenomena leading to predictive models of these phenomena' (Council, 2005). It appeared for the first time in 1736, when the mathematician Leonhard Euler solved the notable Königsberg Bridges problem using graph theory. However, network science acquired the status of scientific discipline on its own in the late 1990s, when scientists in various fields like physics, technology, sociology, biology started to use networks for modeling different complex systems. From there on, the paradigm 'networks are everywhere' became routine and nowadays there are networks in many more interdisciplinary research areas such as neuroscience, epidemics, cybersecurity and so on. Although these networks present several differences between each subfield and they are generated with different processes, a key discovery of network science is that the architecture of networks emerging in various domains is similar; this is a consequence of having analogous organizing principles. Consequently, a common set of mathematical tools is required to explore these systems (Barabási & Pósfai, 2016).

Networks are graphs that represent interactions among individual elements: their structure is given by a collection of nodes and links that describe some kind of relationships between them[1]. This abstract representation offers a common tool to study systems that may differ greatly in nature, appearance, or scope. Nonetheless, real-world data are often subject to greater complexity due to additional information, both in the structural and compositional dimension as well as the existence of multiple interactions among nodes. This has given rise to more complex structures, named attributed and multilayer networks.

Among all the useful concepts for network data analysis, community detection

[1]In this work, the terms graph and network, vertex and node, as well as edge and link are used interchangeably.

arose in the latest years as one of the most popular and studied problem in this context. It consists in partitioning vertices into densely connected components, also known as communities. These share common properties and/or play similar roles within the graph. Real and concrete applications can be found for instance in market segmentation, parallel computing, recommendation systems and many more.

In this work, we propose a new approach for clustering multilayer and attributed networks, which captures the emergent behaviour of complex systems. The goal is to assign each network node (shared across network layers) to clusters, considering altogether the extra information carried by nodes and the connectivity patterns in each layer. This is a challenging task because one has to combine two types of information (Yang et al., 2013), while leveraging the extent to which topological and attribute information contribute to the network's partition (Falih et al., 2018). We present an extension of the existing *MultiTensor (MT)* model recently developed by De Bacco et al. (2017), which performs an overlapping community detection task on multilayer networks by taking into account the interactions among the system's constituents. Specifically, we describe *MultiTensorCov (MTCov)*: this model considers both sources of information for uncovering groups of nodes that are structurally close but also share some common characteristics.

The thesis is structured as follows:

- Chapter 1 reviews the classical literature on community detection and summarizes the main approaches on community detection analysis with node attributes, both on single and multilayer networks.
- Chapter 2 presents the baseline *MultiTensor* model for multilayer community detection, together with its extension *MultiTensorCov*. It defines the main assumptions of our approach and provides the mathematical details, as well as a description of the Expectation-Maximization algorithm used for performing inference.
- Chapter 3 describes an application to a real case study. It presents and summarizes the characteristics of three real social networks, as well as their analysis and results. We provide both quantitative and qualitative results, comparing *MTCov* with its restricted version equivalent to standard *MultiTensor*, i.e. without using extra information of node attributes.
- Chapter 4 is a discussion about conclusions and future works.

1. Related Work

In this chapter, we present a brief review of the literature related to the community detection problem. We start by describing the classical approaches following Fortunato (2010). They encompass graph partitioning, hierarchical and partitional clustering methods, divisive and modularity-based algorithms. Those approaches address the community detection problem focusing only on the topology structure, described by the set of nodes and their interactions. In general, one can find many works of this kind in the literature. However, we only cover the broad perspectives highlighting the aspects used in the implementation of our model. For the interested reader, we refer to the reviews mentioned in section 1.1.
Section 1.2 is dedicated to the methods developed for detecting communities on attributed networks, whose nodes are associated with covariates which explain their features. Firstly, we explain how scientists have handled the problem of combining the network structure and the node information in single networks. Then, we present the importance of multilayer attributed networks and few algorithms developed in this framework. We conclude by describing the main characteristics of *MultiTensorCov* model.

1.1 Community Detection: Classical Approach

Many real-world networks display a community structure organization. Discovering this hidden partition may offer insights on how the network is organized. This is of great importance in many disciplines and it has been used in different applications, including identifying fraud in telecommunications networks (Hoffmann et al., 2001), homology in genetic similarity networks (Haggerty et al., 2013) and relationships in social networks (Bedi & Sharma, 2016). Although over the past few years a large number of scientists have studied the community detection problem, each from a different point of view, this remains an ill-defined task. Many approaches to community detection exist, spanning not only different algorithms

and partitioning strategies but also fundamentally different definitions of what it means to be a 'community' (Peel et al., 2017). This diversity is a strength, because networks generated by different processes and phenomena should not necessarily be expected to be well described by the same structural principles. However, it is sometimes necessary to have a guide for handling the community detection problem and many reviews have been written with this aim. For instance, Fortunato and Hric (2016) offer a guided tour through the main aspects of the problem, where they point out strengths and weaknesses of popular methods giving directions to their use. A similar work can be found in Schaeffer (2007), where they discuss the task of identifying a cluster for a specific seed vertex by means of local computation. Instead, Coscia et al. (2011) provided a classification of the existing algorithms by considering their reference definition of community.

Communities (also known as groups, blocks, modules or clusters) are subsets of vertices usually densely connected which share some common properties. According to Fortunato (2010) the traditional methods for their identification are graph partitioning, hierarchical and partitional clustering. The former follows a cut-based perspective, which consists in dividing the vertices in C groups of predefined sizes, such that the number of edges between the groups (cut) is minimal. This category includes not only the classical Kernighan-Lin algorithm (Kernighan & Lin, 1970), but also procedures which minimize measures affine to the cut size such as conductance, ratio cut and normalized ratio cut. Furthermore, spectral clustering is also connected to the cut-based problem formulations, because it can be seen as the relaxation of the original, combinatorially hard, discrete optimisation problem (Rosvall et al., 2017). It is a technique based on using the eigenvectors of matrices; we refer to Von Luxburg (2007) for a thorough tutorial on this topic. Hierarchical clustering is instead used when the network has several levels of grouping of the vertices. In this case, the algorithms take in input a similarity matrix and create communities including the most similar vertices, either following an agglomerative or a divisive way. They are closely related to the selected similarity measure and the hierarchical structure is sometimes a strong assumption. The partitional clustering is the last traditional method and it includes one of the most popular and oldest algorithm: K-means clustering. In this class, a dissimilarity measure is assigned to each pair of points and they are separated in C clusters such as to maximize/minimize a cost function based on distances between points and/or from points to centroids (Fortunato, 2010). However, the solution found may not be optimal. In addition, as in graph partitioning, the number of clusters has to be

given in input a priori.

Historically later, the divisive algorithms have been developed and the most important example is the Girvan and Newman (2002) model. Here, communities are created by removing edges iteratively based on edge-betweenness score. The general idea is to look for those edges which connect the communities and remove them. The particular choice of centrality measure is the main difference between the techniques in this family. Girvan and Newman have also made another important contribution to the literature of community detection: the introduction of the modularity measure (Newman & Girvan, 2004). This is used both as quality measure for the clustering and as cost function to be optimized. This is still widely used thanks to a huge number of different methods based on this quantity. However, it has also some limits, well described by Lancichinetti and Fortunato (2011).

Although the mentioned methods are quite popular, they do not provide general and viable tools. Indeed, they often return a local optimum and have problems with the size of the networks, as well as with the size of the communities. Moreover, the majority of those methods depends on the choice of the similarity measure and they are often unable to find *overlapping* communities, as each vertex is constrained to belong to a single group. Furthermore, those algorithms force the communities to have an assortative structure, i.e. the probability of connection between nodes of the same community is higher than the probability that nodes in different communities are connected. However, real-world networks often present also disassortative, multipartite and core-periphery structure (Fortunato & Hric, 2016). It is thus important that an algorithm is flexible in capturing all of them.

A more general framework can be found in methods based on statistical inference, which offers tools such as *generative models*. They provide a probabilistic approach for modeling how the network might have been generated, and some of the model parameters, also called community membership vectors, reflect the communities in the network. The community memberships of each node are inferred by fitting the model to the network data (Wang et al., 2013) by using maximum likelihood optimization. This approach is one of the most powerful among those for community detection, since it does not impose any constraint about the structure, it allows to perform inference tasks as link prediction and gives the possibility to assign nodes to more than one cluster, i.e. overlapping communities. They are increasingly present in real-networks and look for them is much more computa-

tionally demanding. Indeed, looking for overlapping communities means assign to each node a membership value for each cluster, and this increases the number of parameters. Algorithms for performing this task have been developed and they have contributed to the expansion of the literature on community detection. A popular review focused on this task has been published by Xie et al. (2013), which described also some generative models. Another feature that may not be handled well by traditional methods is the presence of directed links. In fact, different extensions to the above mentioned models have been developed to incorporate this characteristic. They are well explained by Malliaros and Vazirgiannis (2013). Also in this case, generative models are effective and widely used. The complexity of real-world networks is also expressed by the presence of different type of interactions among nodes; such networks are named *multilayer networks*. Nowadays, they are everywhere and applications can be found in social sciences, biology and engineering to name a few (Boccaletti et al., 2014). It is then important to take multiple types of edges into account if one wants to improve the understanding of complex systems (Kivelä et al., 2014). In order to find communities in these particular graphs, algorithms based on quality functions such as modularity (Mucha et al., 2010) have been considered, as well as those belonging to the hierarchical clustering class (Liu et al., 2018). However, also in this case, the most recent methods are based on fitting generative models as the one developed by De Bacco et al. (2017). This model is flexible as it does not assume *a priori* any particular network structure and it handles overlapping communities, directed and weighted networks in a unified way. Since these properties capture the complexity of real-world networks, we adopt as a starting point the *MultiTensor* algorithm presented in De Bacco et al. (2017).

1.2 Community Detection on Attributed Networks

The methods presented in the previous section handle the community detection task using only the topological structure, i.e. the set of edges connecting the nodes. However, most real-networks data are often associated with additional information, i.e. vertices of a graph are linked with a number of attributes that describe the vertex (Falih et al., 2018). We call these networks *attributed networks*. The community partition can in principle be influenced by both the network topology and the node attributes, thus it is important to consider both sources of information simultaneously and consider network communities as sets of nodes that are densely connected while sharing some common attributes (Yang et al., 2013).

Recently, several studies have combined the structural and compositional dimensions addressing the problem of clustering in attributed networks. The structural dimension refers to the interactions between the nodes; the compositional one concerns the features of the nodes. A comprehensive survey about the main existing clustering methods highlighting their conceptual differences has been provided by Bothorel et al. (2015).

1.2.1 Single Layer Attributed Networks

The majority of the existing algorithms for detecting communities in an attributed network considers only the single layer case, i.e. graphs where at most one type of edge is allowed between two nodes. Through this section we present some examples, underlying different approaches used to handle this problem.

Yang et al. (2013) proposed the Communities from Edge Structure and Node Attribute (CESNA) algorithm, where communities and their attributes are simultaneously detected in an efficient manner. It is a probabilistic generative model which models the links of the network and the node attributes with two independent Bernoulli distributions. They are combined with a unique log-likelihood including a regularization parameter and it is optimized by block-coordinate ascent method. They assumed that communities generate both the network as well as attributes, allowing a dependence between these two sources of information. CESNA algorithm is designed for detecting overlapping communities and it has been proved to be more accurate, more robust, faster and able to work with larger networks than the previous models developed in this field. Moreover, they proved the importance of including the node covariates especially in noisy and not-fully observed networks. In addition, a better interpretability of the detected communities is achieved by finding relevant attributes for each group.

Newman and Clauset (2016) presented a mathematically principled approach to combine the two sources of information. Their method does not assume that the covariates are correlated with the communities. Instead, the algorithm detects and quantifies the agreement between attributes and clusters, if one exists. Otherwise, the node information is ignored. The correlation between the compositional and the structural dimension is a helpful measure both for classifying nodes with missing data and for improving the accuracy. They used a Bayesian statistical inference technique for developing a generative model which modifies the stochastic block model (Holland et al., 1983), that includes a degree-correction term and

a dependence on node attributes through a set of prior probabilities. Moreover, they implemented a belief propagation scheme to perform inference tasks faster and scalable to large networks. In contrast to CESNA, which considers only binary attributes, their algorithm can take in input both numerical and categorical values. Moreover, this method does not assume *a priori* any particular structure for the communities, i.e. assortative/disassortative, and allows to select between competing divisions of a network.

An approach in line with the previous ones is the joint model for data and attributes developed by Hric et al. (2016). They described the network structure and the attributes as a single graph with two layers and developed a non parametric Bayesian inference method that requires no prior information, such as the number of communities. This algorithm is able to quantify the extent to which the covariates are related to the network structure, and vice versa. However, in addition they assessed the attributes in their power to predict the network structure, instead of simply measuring correlation with latent partitions. Furthermore, they showed how the connection between network structure and annotations can be used to predict missing nodes in networks where only the annotations are known, as well as predicting covariates.

Similar works can be found in Xu et al. (2012) and He et al. (2017), where other Bayesian generative models are presented. The former proposed a model-based approach and formulated the community detection as a probabilistic inference problem solved by devising a variational approach and designing an efficient approximate algorithm. The latter described a method for jointly identifying communities and deriving also their semantic description. In this case the model is trained using a method which combines a nested expectation-maximization algorithm and a belief propagation process (NEMBP). Furthermore, Bayesian generative models for attributed graph clustering have been also developed in topic modeling applications; an example is given by Zhu et al. (2013).

Up to this point we only mentioned methods based on statistical inference, however, we can find other approaches in the literature. For instance, Zhang et al. (2016) proposed a joint criterion for community detection (JCDC) with node attributes, which could be seen as a covariate-reweighted modularity. Binkiewicz et al. (2017) used node covariates to help uncover latent communities in a graph using a modification of spectral clustering; whereas Tang and Ding (2019) showed that

the joint clustering problem can be formulated as a spectral relaxation problem.

In summary, although the mentioned works belong to different classes of methods (graph partitioning, hierarchical clustering, generative models via Bayesian inference or likelihood based) and they use different methodologies, they all show that using attributes to detect communities improves the performances of the algorithms with respect to the classical approaches. In addition, they are more scalable and more robust in the presence of noise in the network structure and/or in the attributes. In addition, they provide better results' interpretability. However, we remark that they all consider only single layer networks.

1.2.2 Multilayer Attributed Networks

A more flexible representation of real-world networks is given by multilayer attributed networks. These are networks where nodes have attributes and are shared across layers, each one describing a different interaction between them. Here, we focus on the community detection problem on this kind of networks since the majority of the methods developed for detecting communities in annotated networks, so far only uses a single layer. Given that all of them show prediction performance improvement compared to the methods which consider only one of the two modalities, we expect that using both information would help also in the multilayer case. Despite the extension to multilayer networks has been mentioned as 'future work' in more than one paper, the actual implementation of this extension is still missing. In fact, the literature covering this problem is quite limited.

Recently, Gheche et al. (2018) developed the OrthoNet model which considers together the network information (with multiple layers) and the node features. It is a two-step algorithm: first, it aggregates the layers into a graph representation given by the geometric mean of the network Laplacian matrices; second, it uses a neural network trained by stochastic gradient descent to learn a feature embedding that is consistent with the multilayer structure. Works in line with this method are given by Dong et al. (2013), Chen and Hero (2017) and Mercado et al. (2018), which differ from OrthoNet in the aggregation step. These novel models perform quite well in Purity, Normalized Mutual Information, and adjusted Rand Index measures, both on synthetic and real datasets. However, they follow a neural network approach which is quite different from the one we take here, which is based on statistical inference.

To the best of our knowledge, *MultiTensorCov* is the first model which tries to solve a community detection task joining the multilayer structure and the information carried by the nodes within a probabilistic approach. Our model belongs to the statistical inference methods and benefits from all their properties, from the modeling of *undirected/directed networks*, to the identification of *overlapping communities*. Moreover, it has been designed so to capture the greater complexity of real-world data given by the simultaneous existence of different type of interactions among nodes (*multilayer networks*) as well as the presence of attributes (*attributed networks*). In addition, our model allows the co-existence of different community structures for each layer, including arbitrary mixtures of *assortative, disassortative and core-periphery structure*. Finally, inference tasks as *link prediction* can be performed. The model is explained in detail in Chapter 2.

2. The Model

In this chapter, we present the *MultiTensorCov* model developed for detecting communities in multilayer annotated networks. As a starting point, we fix the notation that will be used along this thesis and then we proceed describing the algorithm. Section 2.2 is dedicated to a brief review of the *MultiTensor (MT)* model. Following the approach developed by De Bacco et al. (2017), we summarize the motivations behind such method, highlighting its properties and mentioning some of the main results. Section 2.3 covers the core of the thesis, describing in depth the mathematical model developed for the community detection task on multilayer networks with node attributes. We present *MultiTensorCov (MTCov)*, which extends the *MT* model incorporating the nodes' attributes and assuming dependency between communities and attributes. Following the formalism of maximum likelihood estimation, the idea underlying this approach is to combine the structural and the node information into a global likelihood function and provide a highly scalable Expectation-Maximization algorithm for estimating the parameters.

2.1 Notation

A multilayer network can be represented as a multilayer graph

$$\mathcal{G} = \{\mathcal{G}^{(\alpha)}\left(\mathcal{V}, \mathcal{E}^{(\alpha)}\right)\}_{1 \leq \alpha \leq L} \tag{2.1}$$

defined on a set $\mathcal{V}$ of N vertices shared across $L \geq 1$ layers. For each layer $\alpha \in \{1, \ldots, L\}$, there is a graph $\mathcal{G}^{(\alpha)}(\mathcal{V}, \mathcal{E}^{(\alpha)})$ and we denote by $A^{(\alpha)} = [a_{ij}^{(\alpha)}] \in \mathbb{R}^{N \times N}$ its adjacency matrix[1]. In addition, for each node $i \in \{1, \ldots, N\}$ let $X_i \in \mathbb{R}^{1 \times K}$ be its features vector, where K is the total number of attributes. Here, we consider only $K = 1$ and in particular, the case where the attribute is categorical with Z different categories.

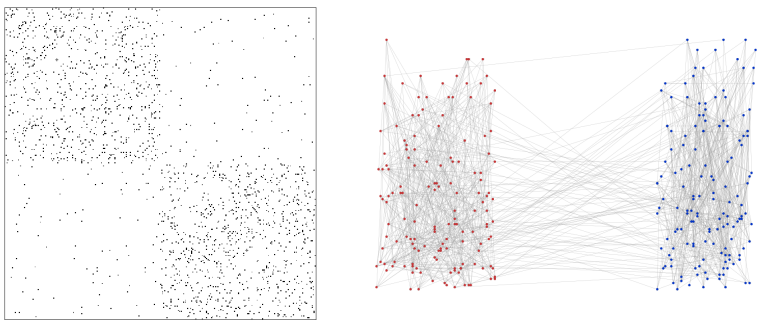
[1] In this work, we consider only positive discrete numbers for the entries of the adjacency matrices.

A community is a subset of vertices that share some properties and the aim of *MTCov* is to find C overlapping communities using the information in the adjacency tensor $A = \{A^{(\alpha)}\}_{1 \leq \alpha \leq L}$ and the design matrix $X = \{X_i\}_{i \in \{1,\dots,N\}}$. In this contest, each node can belong to more than one group. Since we are interested in directed networks, for each node i we have two membership vectors, u_i and v_i. These determine how i forms outgoing and incoming links respectively. For undirected networks, we set $u_i = v_i \, \forall i$. Moreover, each layer α has an affinity matrix $W^{(\alpha)} = [w_{kl}^{(\alpha)}] \in \mathbb{R}^{C \times C}$ which describes the density of edges between each pair (k, l) of groups. Furthermore, each community $k \in \{1, \dots, C\}$ is linked to a category $z \in \{1, \dots, Z\}$ by a parameter β_{kz}, that explains how much information of the z-th category is used for creating the k-th community.

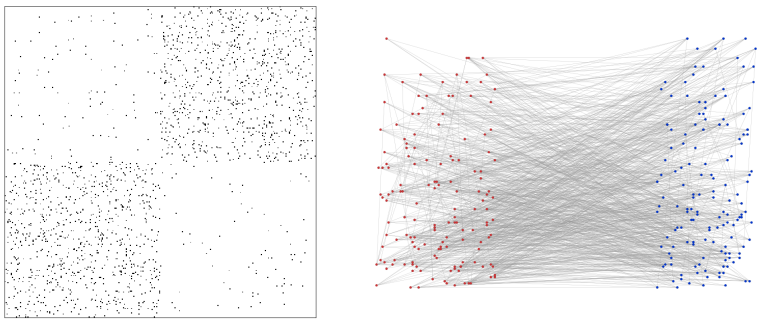
2.2 MultiTensor

Recently, De Bacco et al. (2017) presented the *MultiTensor (MT)* approach in a paper titled 'Community detection, link prediction, and layer interdependence in multilayer networks', which extends and generalizes the mixed-membership stochastic block model (Airoldi et al., 2009). Such approach belongs to the category of generative models and allows to solve three inference problems taking as input the structural dimension: community detection, link prediction and layer interdependence. Both directed and undirected networks can be analyzed with this model and it yields a 'mixed-membership' partition, where each node may belong to more than one community. Although this partition is shared by all layers, *MT* allows patterns in each of them, including arbitrarily mixtures of assortative, disassortative, and core-periphery structure. An example of network layers with different community-relationships is given in Figure 2.1. It shows the adjacency matrices and the representative realizations of a synthetic network with three mixed-membership layers and two communities. In addition, the *MultiTensor* method gives a principled framework for performing link prediction, which is a particularly relevant task when considering real-world networks, as they are often noisy. Moreover, the authors show how to use this information for measuring layer interdependence, a useful knowledge for discarding redundant layers as well as a hint about causal or structural relationships between the layers.

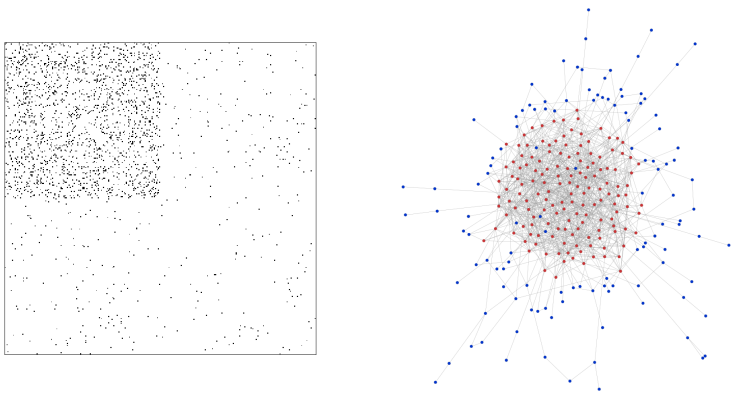
MultiTensor generates multilayer networks (as the one described in eq. (2.1)) assuming an underlying structure that consists of C overlapping communities. It



(a) Assortative structure.



(b) Disassortative structure.



(c) Core-periphery structure.

Figure 2.1: (Left) Adjacency matrices and (right) representative realizations for each layer (a)-(c) of the synthetic network generated by using the stochastic block model. The network has three layers with different structure (assortative, disassortative and core-periphery) and two communities colored in blue and red.

infers the two membership matrices $U, V \in \mathbb{R}^{N\times C}$ assuming non-negative entries and without performing any normalization, as well as for the affinity tensor $W \in \mathbb{R}^{C\times C\times L}$. Then, it models independently each entry of the adjacency tensor $A \in \mathbb{R}^{N\times N\times L}$ with a Poisson distribution, i.e.

$$a_{ij}^{(\alpha)} \sim Pois(M_{ij}^{(\alpha)}) \quad , \tag{2.2}$$

where

$$M_{ij}^{(\alpha)} = \sum_{k,l=1}^{C} u_{ik} v_{jl} w_{kl}^{(\alpha)}. \tag{2.3}$$

Let Θ be shorthand for all $2NC + C^2L$ parameters (given by $u_{ik}, v_{jl}, w_{kl}^{\alpha}$) and assume that all are equally likely *a priori*, then

$$P(\Theta|A) \propto P(A|\Theta) = \prod_{i,j=1}^{N} \prod_{\alpha=1}^{L} \frac{e^{-M_{ij}^{(\alpha)}} \left(M_{ij}^{(\alpha)}\right)^{A_{ij}^{(\alpha)}}}{A_{ij}^{(\alpha)}!} \quad , \tag{2.4}$$

and the log-likelihood is

$$\mathcal{L}(\Theta) = \sum_{i,j,\alpha} \left[A_{ij}^{(\alpha)} \log \sum_{k,l} u_{ik} v_{jl} w_{kl}^{(\alpha)} - \sum_{k,l} u_{ik} v_{jl} w_{kl}^{(\alpha)} \right]. \tag{2.5}$$

They provide an expectation-maximization algorithm in order to maximize the log-likelihood in eq. (2.5) and find estimates for the Θ parameters. Its running time per iteration is linear in the total size of the dataset and it converges quickly in practice. We explain this variational approach in more details throughout the next section.

De Bacco et al. analyzed both synthetic and real data showing the good performances of the algorithm in terms of community detection and link prediction. They used multilayer benchmarks to assess the ability of the model and it achieved the highest cosine similarity compared to the diagonal version of *MT* (which restricts the communities to show assortative structures) and the fully Bayesian Poisson tensor factorization model (Schein et al., 2015). Moreover, they generated synthetic networks with different types of structure in different layers, as the one showed in Figure 2.1. In this contest, *MT* achieved significantly greater performance than the other two methods, highlighting its flexibility in modelling mixtures of connectivity patterns. In addition, the layer interdependence task

via link prediction has been illustrated by analyzing two real-world datasets, one representing social relationships among Indian villagers and the other describing sequence sharing among malaria's virulence genes. In both cases, the *MultiTensor* approach provided information not revealed in previous studies of the two datasets, and proved to be useful in identifying both the presence and the absence of a meaningful structure.

The flexibility of this model, together with its capability of well describing the complexity of real-world networks, make *MultiTensor* a valid and general baseline to start with.

2.3 MultiTensorCov

A straightforward extension of the *MultiTensor* model presented above is given by considering both sources of information carried by a complex network. Here, we present *MultiTensorCov (MTCov)*, an algorithm for community detection in multilayer attributed networks.

We describe separately the procedures for modelling the topology of the network and the node attributes and then we show how to combine them in a unified log-likelihood framework. *MultiTensorCov (MTCov)* generates the network and the attribute probabilistically, assuming an underlying structure consisting of C overlapping communities. Given an observed multilayer network with adjacency tensor A and design matrix X, our goal is to simultaneously infer the node's membership vectors u_i and v_i $\forall i \in \{1, ..., N\}$; the affinity matrices $W^{(\alpha)}$ $\forall \alpha \in \{1, \dots, L\}$, and the $\beta = [\beta_{kz}] \in \mathbb{R}^{C \times Z}$ parameters for the correlation between communities and attributes.

2.3.1 Modeling the links of the network

Following the *MultiTensor* approach and equations (2.2)-(2.5), for each i, j, α we choose $a_{ij}^{(\alpha)}$ independently from the Poisson distribution with mean $M_{ij}^{(\alpha)}$, where

$$M_{ij}^{(\alpha)} = \sum_{k,l=1}^{C} u_{ik} v_{jl} w_{kl}^{(\alpha)}. \tag{2.6}$$

This means that the tensor $M \in \mathbb{R}^{N \times N \times L}$ contains the λ parameters of the NNL Poisson distributions, which are the generative models for the entries of the adja-

cency tensor A. Due to their independence, we can write

$$\mathbb{P}(A|U,V,W) = \prod_{i,j=1}^{N}\prod_{\alpha=1}^{L} \frac{e^{-M_{ij}^{(\alpha)}}\left(M_{ij}^{(\alpha)}\right)^{A_{ij}^{(\alpha)}}}{A_{ij}^{(\alpha)}!}. \tag{2.7}$$

The log-likelihood for the structural dimension ($\mathcal{L}_{\mathcal{G}}$) is then

$$\begin{aligned}\mathcal{L}_{\mathcal{G}} &= \sum_{i,j,\alpha}\left[A_{ij}^{(\alpha)}\log\sum_{k,l}M_{ij}^{(\alpha)} - \sum_{k,l}M_{ij}^{(\alpha)}\right] \\ &= \sum_{i,j,\alpha}\left[A_{ij}^{(\alpha)}\log\sum_{k,l}u_{ik}v_{jl}w_{kl}^{(\alpha)} - \sum_{k,l}u_{ik}v_{jl}w_{kl}^{(\alpha)}\right].\end{aligned} \tag{2.8}$$

2.3.2 Modeling the node attribute

In this work, we consider only one categorical attribute, and for each node the value of the attribute X_i is described by a Multinomial distribution. When the number of categories is given by Z, the response of the i-th node can be codified by using a one hot encoding technique as

$$x_i = (x_{i1}, \ldots, x_{iZ}), \tag{2.9}$$

where $x_{iz} = 1$ if it is observed the z-th category and 0 otherwise. In this way, the original design matrix $X_{N\times 1}$ is translated into a binary matrix $X_{N\times Z}$. Moreover, for each node, $\sum_{z=1}^{Z} x_{iz} = 1$ and x_i can be taken as realization of the random variable

$$X_i = (X_{i1}, \ldots, X_{iZ}) \sim Multi(1, \pi_i). \tag{2.10}$$

The probability function is written as

$$\mathbb{P}(X_i = x_i) = \mathbb{P}(X_{i1} = x_{i1}, \ldots, X_{iZ} = x_{iZ}) = \pi_{i1}^{x_{i1}} \ldots \pi_{iZ}^{x_{iZ}} \quad , \tag{2.11}$$

where the parameter π_i is given by

$$\pi_i = (\pi_{i1} \ldots, \pi_{iZ}) \tag{2.12}$$

$$\text{s.t.} \quad \pi_{iz} \in [0,1] \quad \text{and} \quad \sum_{z=1}^{Z}\pi_{iz} = 1, \forall i. \tag{2.13}$$

Furthermore, the z-th component π_{iz} represents the probability that, for the i-th node, it is observed the category z. A hypothesis we made is that, based on a node's community membership, one should be able to predict the value of each node's attribute. Thus, we write

$$\pi_{iz} = \frac{1}{2}\sum_{k=1}^{C} \beta_{kz}(u_{ik} + v_{ik}) \quad , \tag{2.14}$$

where β_{kz} is the probability of observing a particular category z together with a community k. In order to satisfy the sum constraint in eq. (2.13), we impose the following normalisation:

$$\forall k \quad : \quad \sum_{z=1}^{Z} \beta_{kz} = 1 \tag{2.15}$$

$$\forall i \quad : \quad \sum_{k=1}^{C} u_{ik} = 1 \tag{2.16}$$

$$\forall i \quad : \quad \sum_{k=1}^{C} v_{ik} = 1. \tag{2.17}$$

Such restrictions (2.15)-(2.17) are a particular case for which the general constraint for the multinomial parameter is satisfied. Although they are not the only choices, they allow us to give a probabilistic meaning to the components of β, U and V. Since the attribute has independent components, we write the log-likelihood for the compositional dimension ($\mathcal{L}_{\mathcal{X}}$) as

$$\begin{aligned} \mathcal{L}_{\mathcal{X}} &= \sum_{i=1}^{N}\sum_{z=1}^{Z} x_{iz}\log(\pi_{iz}) \\ &= \sum_{i,z} x_{iz}\log\left(\frac{1}{2}\sum_{k}\beta_{kz}(u_{ik}+v_{ik})\right). \end{aligned} \tag{2.18}$$

2.3.3 The log-likelihood and the EM algorithm

In order to put together the two log-likelihoods (2.8) and (2.18), we introduce a scaling parameter $\gamma \in [0, 1]$ that controls the scaling between the two contributions. We are aware that it is a hyperparameter that must be estimated, but its optimal value obtained *via* tuning techniques (for instance cross-validation) provides a measure for the dependence between the communities and the two sources

of information. Thus, the optimization problem we aim to solve is:

$$\begin{aligned}\hat{U},\hat{V},\hat{W},\hat{\beta} &= \underset{U,V,W,\beta}{\operatorname{argmax}}\,\mathcal{L} \\ &= \underset{U,V,W,\beta}{\operatorname{argmax}}\,[\gamma\mathcal{L}_{\mathcal{X}} + (1-\gamma)\mathcal{L}_{\mathcal{G}}] \\ &= \underset{U,V,W,\beta}{\operatorname{argmax}}\,\gamma\Bigg(\sum_{i,z} x_{iz}\log\Bigg(\frac{1}{2}\sum_{k}\beta_{kz}(u_{ik}+v_{ik})\Bigg)\Bigg)+ \\ &+ (1-\gamma)\Bigg(\sum_{i,j,\alpha}\Bigg[A_{ij}^{(\alpha)}\log\sum_{k,l}u_{ik}v_{jl}w_{kl}^{(\alpha)} - \sum_{k,l}u_{ik}v_{jl}w_{kl}^{(\alpha)}\Bigg]\Bigg). \qquad (2.19)\end{aligned}$$

We maximize $\mathcal{L}$ as a function of $\Theta = (U, V, W, \beta)$ parameters using an Expectation-Maximization (EM) algorithm. Following this variational approach, we introduce two probability distributions: h_{ikz} and $\rho_{ijkl}^{(\alpha)}$. For each i, z with $X_{iz} = 1$, h_{izk} represents our estimate of the probability that the i-th node has the z-th category, given that it belongs to the community k. On the other hand, for each i, j, α with $A_{ij}^{(\alpha)} = 1$, $\rho_{ijkl}^{(\alpha)}$ is the probability distribution over pairs of groups k, l. Thus, the EM algorithm consists of randomly initializing the parameters Θ and then by alternatively updating h, ρ and Θ until $\mathcal{L}$ reaches a fixed point.
Using the Jensen's inequality, we write

$$\begin{aligned}\mathcal{L}_{\mathcal{X}} &\geq \sum_{i,z}x_{iz}\sum_{k}h_{izk}\log\frac{\beta_{kz}(u_{ik}+v_{ik})}{2h_{izk}} \\ &= \sum_{i,z,k}x_{iz}(h_{izk}\log(\beta_{kz}(u_{ik}+v_{ik})) - h_{izk}\log(2h_{izk})) \qquad (2.20)\\ \mathcal{L}_{\mathcal{G}} &\geq \sum_{i,j,\alpha}\Bigg[A_{ij}^{(\alpha)}\sum_{k,l}\rho_{ijkl}^{(\alpha)}\log\frac{u_{ik}v_{jl}w_{kl}^{(\alpha)}}{\rho_{ijkl}^{(\alpha)}} - \sum_{k,l}u_{ik}v_{jl}w_{kl}^{(\alpha)}\Bigg] \\ &= \sum_{i,j,k,l,\alpha}\Bigg[A_{ij}^{(\alpha)}\Bigg(\rho_{ijkl}^{(\alpha)}\log(u_{ik}v_{jl}w_{kl}^{(\alpha)}) - \rho_{ijkl}^{(\alpha)}\log(\rho_{ijkl}^{(\alpha)})\Bigg) - u_{ik}v_{jl}w_{kl}^{(\alpha)}\Bigg]. \qquad (2.21)\end{aligned}$$

These lower bounds hold with equality when

$$h_{izk} = \frac{\beta_{kz}(u_{ik}+v_{ik})}{\sum_{k'}\beta_{k'z}(u_{ik'}+v_{ik'})} \quad , \quad \rho_{ijkl}^{(\alpha)} = \frac{u_{ik}v_{jl}w_{kl}^{(\alpha)}}{\sum_{k',l'}u_{ik'}v_{jl'}w_{k'l'}^{(\alpha)}}, \qquad (2.22)$$

giving us the E step of the algorithm. For the M step, we derive update equations for the parameters U, V, W, β by taking the derivatives of the log-likelihood and setting them equal to zero.

The partial derivative with respect to the elements of the affinity matrices $W^{(\alpha)}$ is

given by

$$\frac{\partial \mathcal{L}}{\partial w_{kl}^{(\alpha)}} = (1-\gamma)\frac{\partial \mathcal{L}_{\mathcal{G}}}{\partial w_{kl}^{(\alpha)}} = (1-\gamma)\Bigg(\sum_{i,j}\left[\frac{A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}}{w_{kl}^{(\alpha)}} - u_{ik}v_{jl}\right]\Bigg). \tag{2.23}$$

The valid update when γ is different from 1, is given by setting eq. (2.23) to zero and we obtain

$$w_{kl}^{(\alpha)} = \frac{\sum_{i,j} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}}{\sum_i u_{ik}\sum_j v_{jl}}. \tag{2.24}$$

In order to take the derivative with respect to β_{kz} we need to consider the Lagrange multiplier λ_k because of the constraint in eq. (2.15). Then,

$$\frac{\partial \mathcal{L}}{\partial \beta_{kz}} = \gamma\left(\frac{1}{\beta_{kz}}\sum_i x_{iz}h_{izk}\right) - \lambda_k \tag{2.25}$$

and the equality with zero implies

$$\beta_{kz} = \frac{\gamma}{\lambda_k}\sum_i x_{iz}h_{izk}. \tag{2.26}$$

Using the eq. (2.15), we have

$$\sum_z \frac{\gamma}{\lambda_k}\sum_i x_{iz}h_{izk} = 1 \quad , \tag{2.27}$$

and so

$$\lambda_k = \gamma\sum_{i,z} x_{iz}h_{izk}. \tag{2.28}$$

Plugging (2.28) into (2.26), we obtain the update

$$\beta_{kz} = \frac{\sum_i x_{iz}h_{izk}}{\sum_{i,z} x_{iz}h_{izk}}. \tag{2.29}$$

Focusing the attention on the elements of the matrix U, we first consider that plugging the update for $w_{kl}^{(\alpha)}$ given in eq. (2.24) into the log-likelihood of the

structural dimension in eq. (2.21), the last term becomes a constant. Indeed,

$$
\begin{aligned}
-\sum_{i,j}\sum_{k,l} u_{ik}v_{jl}\frac{\sum_{i,j} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}}{\sum_i u_{ik}\sum_j v_{jl}} &= -\sum_{k,l}\left(\frac{\sum_{i,j} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}}{\sum_i u_{ik}\sum_j v_{jl}}\sum_{i,j} u_{ik}v_{jl}\right) \\
&= -\sum_{k,l}\left(\sum_{i,j} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}\right) \\
&= -\sum_{i,j} A_{ij}^{(\alpha)}\sum_{k,l}\rho_{ijkl}^{(\alpha)} \\
&= -\sum_{i,j} A_{ij}^{(\alpha)} \\
&= -T^(\alpha) \qquad (2.30)
\end{aligned}
$$

since $\sum_{k,l}\rho_{ijkl}^{(\alpha)} = 1$ and $\sum_{i,j} A_{ij}^{(\alpha)}$ is the number of links in layer α when the network is directed (or twice this value in the undirected case). Thus, we can ignore this term when estimating u_{ik}. Using the same strategy for computing the update of β, we compute the Lagrange multiplier δ_i for the constraint in eq. (2.16). Then,

$$
\frac{\partial\mathcal{L}}{\partial u_{ik}} = \frac{1}{u_{ik}}\left(\gamma\sum_z x_{iz}h_{izk} + (1-\gamma)\sum_{j,l,\alpha} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}\right) - \delta_i \qquad (2.31)
$$

and

$$
u_{ik} = \frac{1}{\delta_i}\left(\gamma\sum_z x_{iz}h_{izk} + (1-\gamma)\sum_{j,l,\alpha} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}\right). \qquad (2.32)
$$

Using (2.16), we have

$$
\sum_k \frac{1}{\delta_i}\left(\gamma\sum_z x_{iz}h_{izk} + (1-\gamma)\sum_{j,l,\alpha} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}\right) = 1 \qquad (2.33)
$$

which implies

$$
\begin{aligned}
\delta_i &= \sum_k\left(\gamma\sum_z x_{iz}h_{izk} + (1-\gamma)\sum_{j,l,\alpha} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}\right) \\
&= \gamma + (1-\gamma)\sum_{j,\alpha} A_{ij}^{(\alpha)} \quad , \qquad (2.34)
\end{aligned}
$$

since $\sum_k h_{izk} = 1$, $\sum_{k,l}\rho_{ijkl}^{(\alpha)} = 1$ and $\sum_z x_{iz} = 1$. Plugging the result of eq. (2.34)

into the equality (2.32) we obtain

$$u_{ik} = \frac{\gamma \sum_z x_{iz} h_{izk} + (1-\gamma) \sum_{j,l,\alpha} A_{ij}^{(\alpha)} \rho_{ijkl}^{(\alpha)}}{\gamma + (1-\gamma) \sum_{j,\alpha} A_{ij}^{(\alpha)}}. \tag{2.35}$$

In order to compute the update for V, we fix j and l, rewriting the equation (2.20) as

$$\mathcal{L}_{\mathcal{X}} = \sum_{j,z,l} x_{jz}(h_{jzl} \log(\beta_{lz}(u_{jl} + v_{jl})) - h_{jzl} \log(h_{jzl})). \tag{2.36}$$

Analogous to before, we consider the Lagrange multiplier to satisfy the restriction given in eq. (2.17), and we obtain

$$v_{jl} = \frac{\gamma \sum_z x_{jz} h_{jzl} + (1-\gamma) \sum_{i,k,\alpha} A_{ij}^{(\alpha)} \rho_{ijkl}^{(\alpha)}}{\gamma + (1-\gamma) \sum_{i,\alpha} A_{ij}^{(\alpha)}}. \tag{2.37}$$

In each iteration of the EM algorithm, the parameters U, V, W, β are updated thanks to equations (2.24), (2.29), (2.35) and (2.37), until the log-likelihood $\mathcal{L}$ reaches a fixed point. An application of our model to some real-networks is presented in the next chapter, where we highlight both strengths and weakness of this approach.

3. A Case Study

In this chapter, we present some applications to real-networks, in order to demonstrate the strengths of our model. We show improvements in terms of the evaluation measures compared to the version of the algorithm that does not consider the attributes, and highlight the interpretability of the results. In section 3.1, we start by describing the network data and summarizing their descriptive characteristics. Then, we present the analysis, the main issues and the results. In the last section of this chapter, we provide both quantitative and qualitative conclusions.

3.1 Data

We analyzed three real-networks that describe the social support interactions between residents of two South Indian villages, that are named by pseudonyms 'Te̲npaṭṭi' (TEN) and 'A̲lakāpuram' (ALA), for privacy reasons. The data have been collected by the anthropologist E. A. Power with help from Madurai Kamaraj University and Chella Meenakshi Centre for Education Research and Services, into two rounds: one in 2013 and the other in 2017. The surveys have been approved by Human Subjects Institutional Review Board of Stanford University and of the University of Cincinnati, respectively. More details about datasets and background of the data research can be found in Power (2015) and Power (2017). The social support surveys were conducted with virtually all adult residents of the two villages and they were asked to name those individuals who are their close friends or relatives, who provided them with different types of support (e.g. money loan, advice) and who plays a particular role within the village. Each question about social support and reputation endorsement has been translated in a layer in the respective network. In addition, some metadata were collected, which include features like age, religion, caste, education and so on.

In previous works, Power (2015) and Power (2017) considered the hypothesis that there is a relationship between religion and prosociality, as well as De Bacco et al.

(2017) suggested an interdependence between the attribute caste and the mechanisms driving edge formation, which means the community structure. We based our analysis on the last hypothesis, assuming an underlying dependence between the communities, the multilayer structure and the compositional dimension given by the attribute caste. Raw data have been pre-processed first. We kept only residents and survey respondents who had at least one edge among the layers. Moreover, we considered the networks without self-loops. Then, the resulting directed networks present the characteristics showed in the Table 3.1, and we refer to them as TEN 2013, TEN 2017 and ALA 2017, respectively.

Village	Year	Nodes	Layers	Average Edges	Average Density
Te̲npaṭṭi	2013	362	12	302.65	0.232
Te̲npaṭṭi	2017	346	11	283.15	0.237
A̲lakāpuram	2017	441	11	435.95	0.225

Table 3.1: Network summary statistics for the three adult resident village networks.

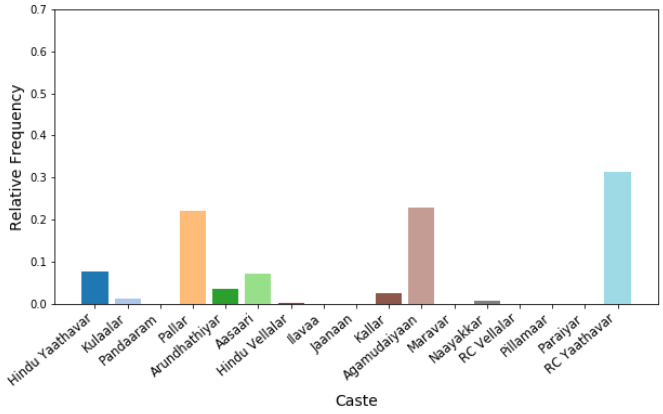
The distribution of the attribute caste inside each network is shown in Figure 3.1. It displays some differences between 'Te̲npaṭṭi' and 'A̲lakāpuram' networks, but also a common unbalanced partition of the categories. This covariate is present with 10 categories in TEN 2013, 9 in TEN 2017 and 13 in ALA 2017.

3.2 Analysis

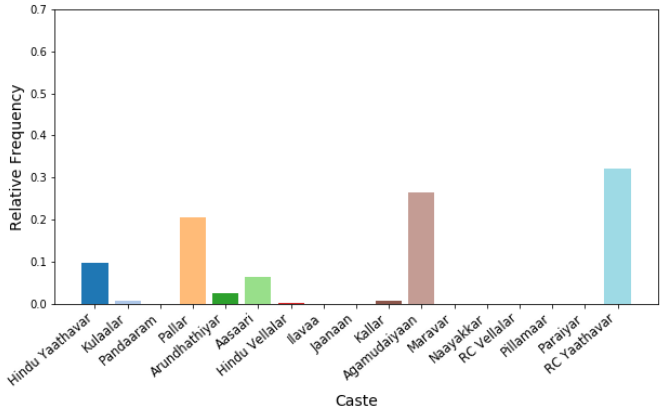
In order to find the communities and measure their interdependence with the covariate, we analyzed the three networks using the *MTCov* model already presented in chapter 2 and implemented in Python and R. We started by investigating the behaviour of the global log-likelihood $\mathcal{L}$, as well as the two parts of which it is composed. Figure 3.2 shows an example for the network ALA 2017 and the attribute caste. The different scaling between $\mathcal{L}_{\mathcal{X}}$ and $\mathcal{L}_{\mathcal{G}}$ graphs is notable, which results in a different contribution on the final log-likelihood. For this reason, we first looked for a normalization procedure, in order to give a reasonable interpretation for the estimated scaling parameter $\hat{\gamma}$.

3.2.1 Normalization

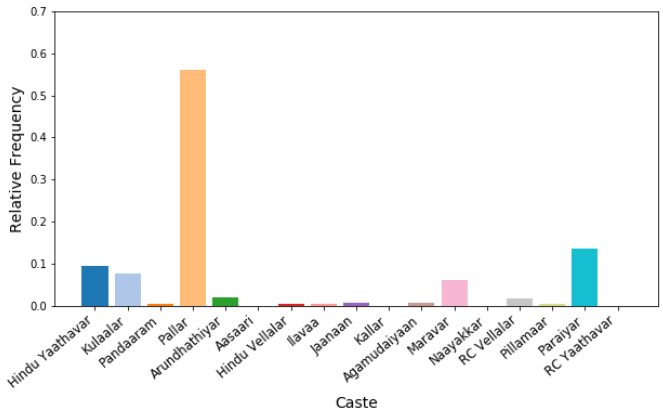
Since the preliminary analysis highlighted a relation between the log-likelihoods, network structures and attributes, we estimated two linear regression models in order to obtain the normalization constants for the two sources of information. We



(a) TEN 2013.



(b) TEN 2017.



(c) ALA 2017.

Figure 3.1: Distribution of the attribute caste inside each network.

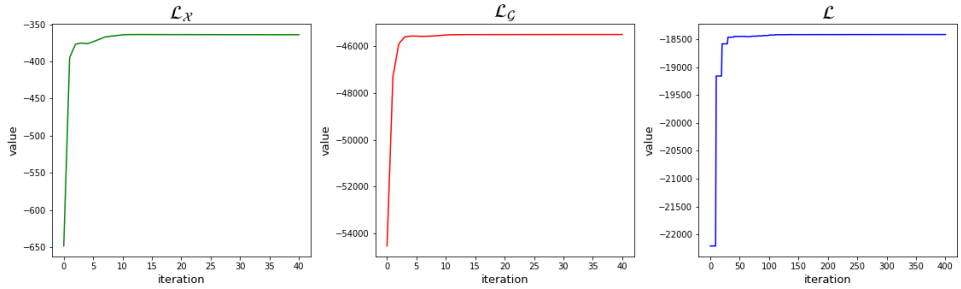


Figure 3.2: Behaviours of the log-likelihood $\mathcal{L}_\mathcal{X}$ (compositional dimension), the log-likelihood $\mathcal{L}_\mathcal{G}$ (structural dimension) and the global log-likelihood $\mathcal{L}$. Here, we report results for the ALA 2017 network and the attribute caste, using $\gamma = 0.6$.

collected log-likelihood values with respect to the number of nodes (N), edges (E) of the observed network and the number of categories of the categorical attribute (Z). Quantitatively, this means normalizing as:

$$\mathcal{L} = \gamma \frac{\mathcal{L}_\mathcal{X}}{c_N^\mathcal{X} N + c_E^\mathcal{X} E + c_Z^\mathcal{X} Z} + (1-\gamma) \frac{\mathcal{L}_\mathcal{G}}{c_N^\mathcal{G} N + c_E^\mathcal{G} E + c_Z^\mathcal{G} Z}. \tag{3.1}$$

The super and the subscripts of the c parameters indicate the dependent variable ($\mathcal{L}_\mathcal{X}$ or $\mathcal{L}_\mathcal{G}$) and the input regressor they refer to, respectively. In particular, we collected the data by running the model for each pair of network and categorical attribute, arbitrarily fixing the number of communities $C = 3$ and the scaling parameter $\gamma = 0.5$. Table 3.2 states the values of the coefficients which resulted statistically significant for the estimation of the log-likelihood terms[1].

	$\mathcal{L}_\mathcal{X}$	$\mathcal{L}_\mathcal{G}$
c_N	-0.486^{***}	-1.778^{**}
c_E		-6.158^{***}
c_Z	-33.862^{***}	

Table 3.2: Coefficient estimates $c_N^\mathcal{X}$, $c_E^\mathcal{X}$, $c_Z^\mathcal{X}$, $c_N^\mathcal{G}$, $c_E^\mathcal{G}$ and $c_Z^\mathcal{G}$ for the two linear regression models.

On one side, this procedure presented coefficient estimates useful for analyzing all three networks as automatically as possible, used in the algorithm for normalizing the respective log-likelihood terms before joining them. On the other side, we are aware that these coefficients are strictly related to the type of network we are working with. In this sense, future works should investigate an automatic normalization procedure for any network.

[1]Only the statistically significant coefficients have been used in the normalized eq. (3.1).

3.2.2 Evaluation measures

Given that we are considering attributed networks without ground-truth about community assignments, model evaluation is inherently challenging. Moreover, in this case we have an additional aspect to consider: the co-existence of multiple objective functions. In fact, we want to identify sets of nodes making a good cluster with respect to all these aspects (Bothorel et al., 2015). For this reason, we decided to define two different evaluation measures, one for the structure and the other for the attribute.

Even though there have been many recent developments in the literature about methods for evaluating graph clustering algorithms that rely on network structure, this is still an open problem as there is no general agreement about what are the best practices. For more details about existing methods, we refer the reader to the following works: Fortunato (2010), Leskovec et al. (2010), Bothorel et al. (2015) and Chakraborty et al. (2017).
Here, given the similarity of applications and contexts, we follow the approach used by Clauset et al. (2008). In particular, we developed a generative algorithm that allows the use of a link-prediction technique both for performing a hyperparameter tuning and for quantifying the accuracy of the predicted partitions. More specifically, we used the AUC statistic, equivalent to the area under the receiver-operating characteristic (ROC) curve (Hanley & McNeil, 1982). It represents the probability that a randomly chosen missing connection (a true positive) is given a higher score by our method than a randomly chosen pair of unconnected vertices (a true negative). Thus, an AUC statistic equal to 0.5 indicates random chance, while the closer it is to 1, the more our model's predictions are better. A strong positive correlation between the community division and the link prediction has been confirmed also by Wu et al. (2018).

For the attribute, instead, we used the accuracy as a quality measure. For each node, we computed the multinomial parameter π_i using eq. (2.14), given the estimated parameters $\hat{\beta}$ and the estimated community membership matrices $\hat{U}$ and $\hat{V}$. We then assigned to each node the category with the highest probability, computing the accuracy as a fraction between the corrected classified examples and the total number of nodes. We compared this value both with the random probability and the highest relative frequency in the training set.

3.2.3 Hyperparameter tuning

In order to evaluate our model on the data, we designed a procedure for tuning the hyperparameters, which are defined as parameters that cannot be directly learned from the data, but usually fixed before the actual training process begins. Here, they are represented by the scaling parameter γ and the number of communities C, given the unsupervised context. The optimal values are those who present the best performance of the model, and they have been chosen by using the 5-fold Cross Validation technique in conjunction with the Grid Search strategy. Such approach consists in first splitting the dataset into K folds: $K-1$ of them make the training set, the remaining fold is the test set (or held-out set). Secondly, repeatedly training the model from scratch on the $K-1$ folds of the sample and testing on the held-out fold, and this procedure is done for all combinations of the enumerated hyperparameters. Then, we choose the combination $(\hat{C}, \hat{\gamma})$ that returns the best averaged performance over the cross-validation runs. Although this procedure might be computationally expensive, it gives us the possibility to adapt our choice in order to have a co-existence of good performance for both objective functions. In fact, working with other strategies or optimization procedures would have returned the optimal value but these may not allow to compare *jointly* the two values.

We choose, as range of possible values for C, the integer numbers in $[2, 15]$. In order to avoid a large number of groups, values greater than 15 have not been considered, while the option 1 corresponds to the trivial case without communities. For γ, we considered values in $[0, 1]$, at intervals of equal length of 0.1. Moreover, the dataset has been split as follows:

- For the structure, instead of removing 20% of the nodes or 20% of the links, we hided 20% of the entries of the adjacency matrix, including both zeros and ones, strategy used by De Bacco et al. (2017).
- For the attribute, we hided 20% of the entries of the categorical vector.

In each iteration, we first trained the model giving as input the adjacency and the design matrices with hidden entries, then we tested on the elements of the fold left out. At each step, we computed both AUC and accuracy on the test set. Figure 3.3 illustrates the average results for all combinations of the hyperparameters (C, γ), for the network TEN 2013. The plots of the other networks are attached in Appendix A. The blue and orange horizontal dashed lines represent the best cross-validation values for the two evaluation measures independently, while the black

line displays the pair of selected hyperparameters. We chose as best parameters the pair that returns the closest values to the best for both objective functions.

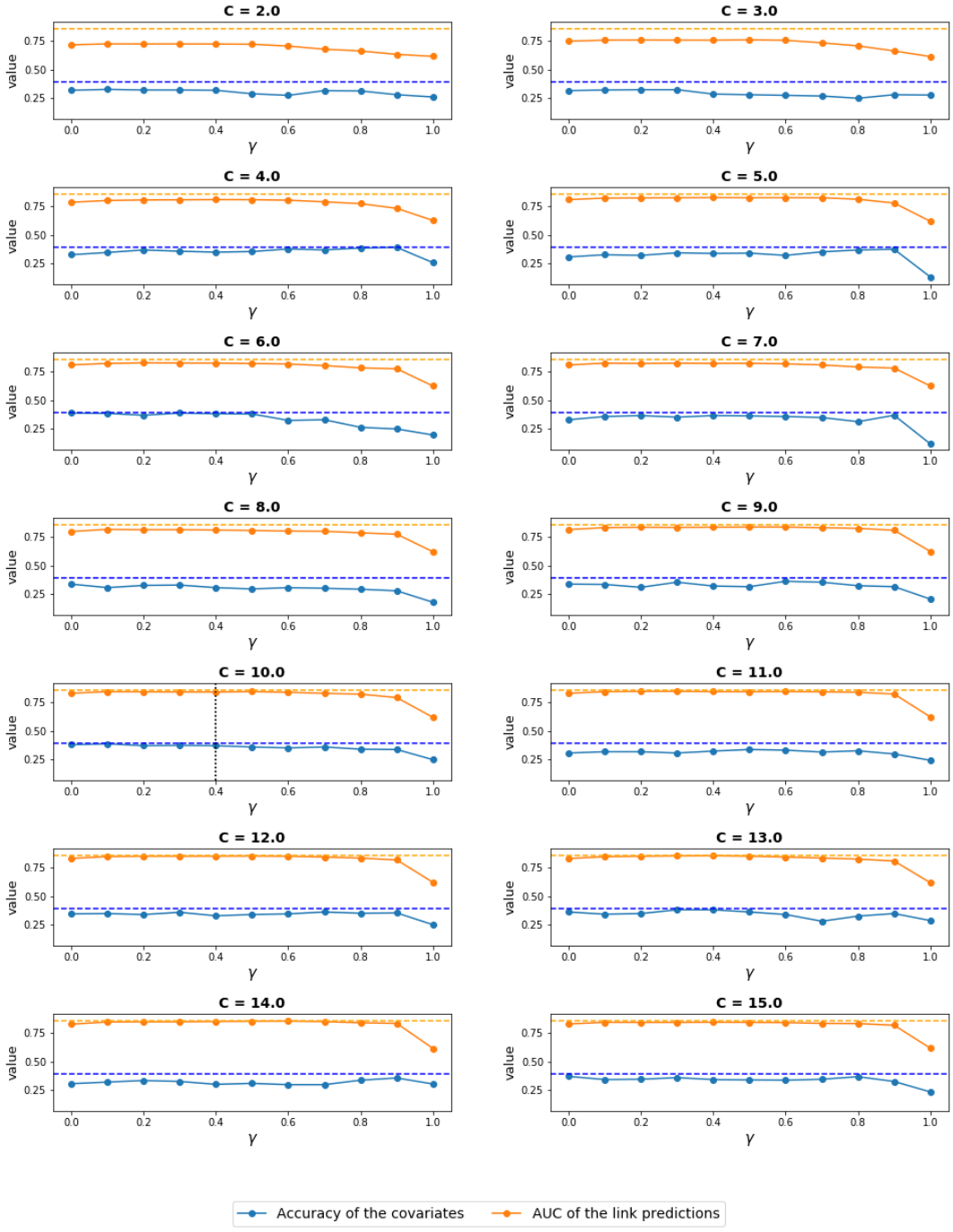


Figure 3.3: 5-fold CV average values for AUC and accuracy for all combinations of the hyperparameters for the network based on Tenpaṭṭi village in the year 2013 and the attribute caste. The blue and orange dashed lines represent the best cross-validation values for the two evaluation measures independently, while the black one displays the pair of selected hyperparameters.

In order to assess if the presence of the covariate helps the algorithm for detecting communities, there are two straightforward models for comparison. One is given by the restricted *MTCov* model with scaling parameter $\gamma = 0$, that allows to take into account only the structural dimension. The other model is represented by *MultiTensor*, which only takes as input the adjacency tensor. However, before starting the validation phase, also the hyperparameter C of these models needed to be set. We used the same 5-fold CV technique, obtaining equivalent final choices for all networks with both models, as showed in Figure 3.4. This was an expected result since the log-likelihood of the *MultiTensorCov* model with $\gamma = 0$ is only composed by the component $\mathcal{L}_{\mathcal{G}}$, which is exactly the objective function of the *MultiTensor* model. Moreover, the parameters updates for the two algorithms are almost the same, up to the normalization constants given by the fact that *MTCov* put some normalization restrictions, while *MT* works in an unconstrained space.

Table 3.3 summarizes the results of the hyperparameter tuning for the three algorithms, and these values are going to be used in the next section for validating the model. It should be noticed that *MTCov* estimated fewer communities, or at most the same number, than the comparison models. This is due to the presence of the scaling parameter γ, that captures the complexity of the data and helps during the formation of communities providing more information. In addition, this parameter ranges between 0.4 and 0.6, meaning that the two sources of information contribute almost equally for detecting the communities. Also, these values show a correlation between the structure, the attribute and the communities.

		MT	**Restricted MTCov**	**MTCov**
TEN 2013	$\hat{C}$	12	12	10
	$\hat{\gamma}$	—	0	0.4
TEN 2017	$\hat{C}$	6	6	6
	$\hat{\gamma}$	—	0	0.5
ALA 2017	$\hat{C}$	10	10	9
	$\hat{\gamma}$	—	0	0.6

Table 3.3: Values of the hyperparameters after tuning, for the three models and networks.

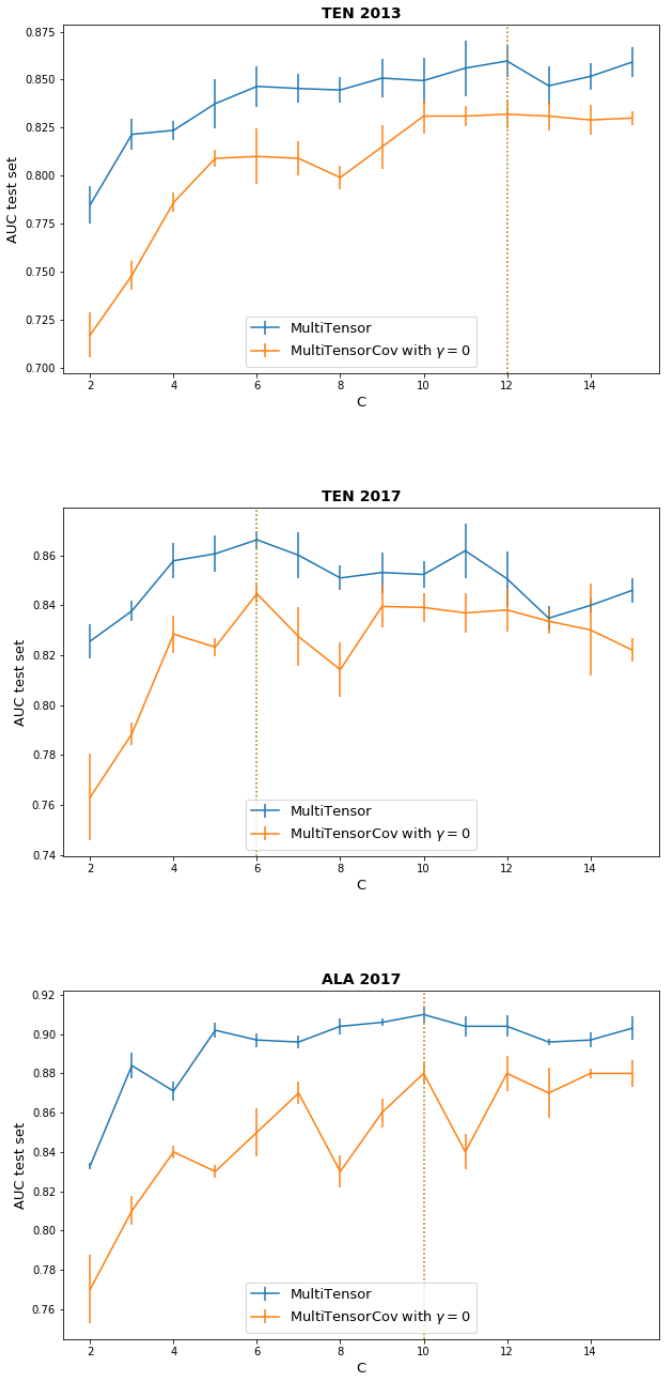


Figure 3.4: 5-fold CV average values for the AUC for all possible choices of the hyperparameter C, considering MT and the restricted $MTCov$ with $\gamma = 0$.

3.3 Results

Once the hyperparameters have been fixed, we assessed the performance of the model by validating its results both in terms of evaluation measures and interpretability of the communities. We analyzed the used networks considering the attribute caste, transformed by one hot encoding technique. Subsection 3.3.1 describes the validation phase and the comparisons with the *MultiTensor* and the restricted *MultiTensorCov* models. Finally, in the subsection 3.3.2 we show the detected communities and present their qualitative interpretations.

3.3.1 Validation

We quantitatively validated the model by performing ten independent runs with different training and test sets, for each network. In each experiment, the datasets have been split as explained in subsection 3.2.3, using the 70% for the training set and the 30% for testing, and we adopted the evaluation measures previously introduced in subsection 3.2.2. For each trial, we estimated the Θ parameters on the training set, used in a second step for computing the AUC and the accuracy on the test set. For comparison, we run the *MultiTensor* algorithm and the restricted version of our model, that considers only the structural part by fixing the scaling parameter $\gamma = 0$. Moreover, we looked at other two baselines for the accuracy, such as the random probability assigned to each category of the attribute (RP) and the maximum relative frequency of the covariate caste in the training set (MRF).

Table 3.4 shows the average results over ten runs for the considered networks, together with their standard deviations. In all cases, our algorithm achieves the highest accuracy for the caste category prediction, and in two cases over three the highest AUC for the edge prediction, indicated in boldface. Nonetheless, even in the single case where our model is not the best, the difference in score is negligible and can be offset by the possibility of inferring the attribute, feature that is distinctive of our method.

To emphasize the importance of the covariate in the community detection task, we implemented experiments with non-random test and training sets. We used sampling bias techniques for assigning higher or lower sampling probability to the entries of the adjacency tensor, which correspond to edges than the non-edges

			TEN 2013		TEN 2017		ALA 2017
ACCURACY for the covariate caste on the test sets	RP		0.1		0.11		0.08
	MRF		0.318 ± 0.014		0.327 ± 0.016		0.555 ± 0.009
	MT	$C = 12$	—	$C = 6$	—	$C = 10$	—
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.317 ± 0.037	$C = 6$ $\gamma = 0$	0.381 ± 0.028	$C = 10$ $\gamma = 0$	0.597 ± 0.028
	MTCov	$C = 10$ $\gamma = 0.4$	**0.342 ± 0.050**	$C = 6$ $\gamma = 0.5$	**0.407 ± 0.033**	$C = 9$ $\gamma = 0.6$	**0.611 ± 0.030**
AUC for the link prediction on the test sets	MT	$C = 12$	**0.830 ± 0.004**	$C = 6$	0.819 ± 0.008	$C = 10$	0.849 ± 0.005
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.809 ± 0.010	$C = 6$ $\gamma = 0$	0.832 ± 0.002	$C = 10$ $\gamma = 0$	0.867 ± 0.006
	MTCov	$C = 10$ $\gamma = 0.4$	0.825 ± 0.012	$C = 6$ $\gamma = 0.5$	**0.840 ± 0.005**	$C = 9$ $\gamma = 0.6$	**0.886 ± 0.005**

Table 3.4: Performance of methods *MT* , restricted *MTCov* and *MTCov* on three social networks. RP is the random probability baseline for the accuracy of the category of caste attribute prediction. MRF indicates the maximum relative frequency linked to the covariate caste in the training set. The results are averaged over ten independent runs and the best outcomes are bolded.

ones. In particular, we assigned to the entries $a_{ij}^{\alpha} = 1$ the probability given by

$$p_1 = \frac{\text{total probability of picking up one edge (tpe)}}{\text{number of edges in the network (E)}} \quad , \tag{3.2}$$

and for 0 entries

$$p_2 = \frac{1 - \text{total probability of picking up one edge (tpe)}}{N - E} \quad , \tag{3.3}$$

where N is the number of the nodes of the network. The tpe was used to select entries for the test set; in case of selecting entries uniformly at random, this value would be around 0.003. This low value is due to the common case in real networks of having sparse matrices, where the number of 1 is much lower than the number of 0. The aim was to assess the importance of having another source of information in situations where the number of positive examples in the training sets is not equally likely as the negative ones.
We created three different situations:

- tpe = 0.002: the probability of selecting one edge is lower than the probability of choosing one non edge, and the number of edges in the training set is much higher than the number in the test set.

- tpe = 0.01: the probability of selecting one edge is higher than the probability of choosing one non edge, and the number of edges in the training set is a

bit smaller than the number in the test set.

- tpe = 0.015: the probability of selecting one edge in the test set is higher than the probability of choosing one non edge, and the test set has a bigger number of positive examples.

The presented frameworks follow an increasing order of complexity, starting from an over-represented case, where tpe = 0.002, and ending with a difficult task where the number of edges in the training set is under-represented, tpe = 0.015. We run 10 independent trials for each setting and model. The performances are shown in Figure 3.5, where the opacity controls the complexity of the frameworks: the lower the value of tpe (less complex), the more transparent the opacity. Moreover, Tables 3.5, 3.6 and 3.7 present the results averaged over the independent runs, obtained in the three settings, respectively.

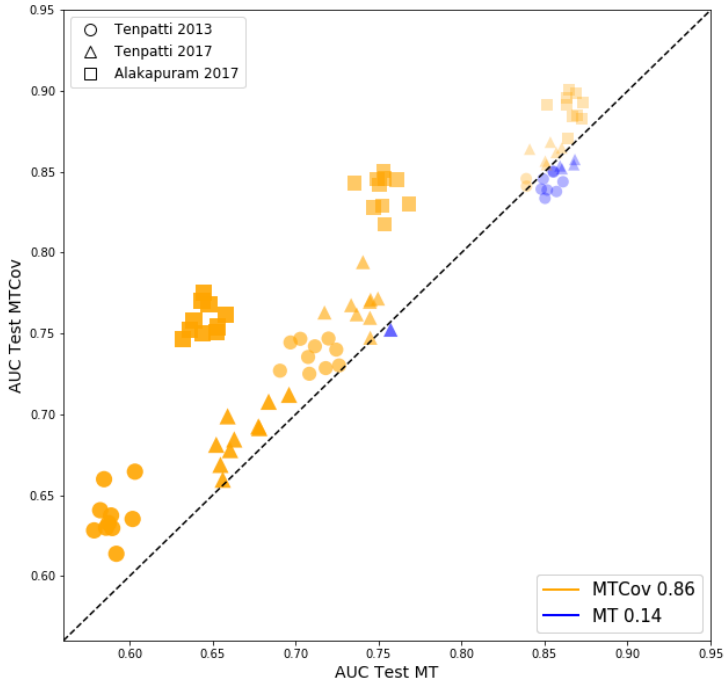


Figure 3.5: Probabilistic link prediction accuracy AUC of *MTCov* and *MT*. The values of AUC for *MTCov* and *MT* are shown on the vertical axis and the horizontal axis respectively. The opacity controls the complexity of the frameworks: on the left tpe = 0.015 (high opacity and complexity), on the center tpe = 0.01 (medium opacity), on the right tpe = 0.002 (low opacity). Points above the diagonal, shown in orange, are trials where *MTCov* is more accurate than *MT*. The fractions for which each method is superior are shown in the plot legend.

			TEN 2013		TEN 2017		ALA 2017
ACCURACY for the covariate caste on the test sets	RP		0.1		0.11		0.08
	MRF		0.314 ± 0.017		0.316 ± 0.010		0.556 ± 0.017
	MT	$C = 12$	—	$C = 6$	—	$C = 10$	—
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.337 ± 0.035	$C = 6$ $\gamma = 0$	**0.376 ± 0.023**	$C = 10$ $\gamma = 0$	0.598 ± 0.036
	MTCov	$C = 10$ $\gamma = 0.4$	**0.364 ± 0.048**	$C = 6$ $\gamma = 0.5$	0.371 ± 0.031	$C = 9$ $\gamma = 0.6$	**0.625 ± 0.034**
AUC for the link prediction on the test sets	MT	$C = 12$	**0.851 ± 0.007**	$C = 6$	0.857 ± 0.008	$C = 10$	0.866 ± 0.006
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.830 ± 0.008	$C = 6$ $\gamma = 0$	0.851 ± 0.007	$C = 10$ $\gamma = 0$	0.879 ± 0.008
	MTCov	$C = 10$ $\gamma = 0.4$	0.843 ± 0.005	$C = 6$ $\gamma = 0.5$	**0.859 ± 0.005**	$C = 9$ $\gamma = 0.6$	**0.889 ± 0.005**

Table 3.5: Setting tpe = 0.002. Performance of methods *MT* , restricted *MTCov* and *MTCov* on three social networks and in a sampling bias framework, where the number of edges in the training set is much higher than the number in the test set. RP is the random probability baseline for the accuracy of the category of caste attribute prediction. MRF indicates the maximum relative frequency linked to the covariate caste in the training set. The results are averaged over ten independent runs and the best outcomes are bolded.

			TEN 2013		TEN 2017		ALA 2017
ACCURACY for the covariate caste on the test sets	RP		0.1		0.11		0.08
	MRF		0.317 ± 0.010		0.317 ± 0.012		0.566 ± 0.014
	MT	$C = 12$	—	$C = 6$	—	$C = 10$	—
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.312 ± 0.019	$C = 6$ $\gamma = 0$	**0.363 ± 0.036**	$C = 10$ $\gamma = 0$	0.564 ± 0.034
	MTCov	$C = 10$ $\gamma = 0.4$	**0.326 ± 0.025**	$C = 6$ $\gamma = 0.5$	0.335 ± 0.037	$C = 9$ $\gamma = 0.6$	**0.613 ± 0.030**
AUC for the link prediction on the test sets	MT	$C = 12$	0.711 ± 0.011	$C = 6$	0.741 ± 0.010	$C = 10$	0.752 ± 0.008
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.698 ± 0.009	$C = 6$ $\gamma = 0$	0.745 ± 0.001	$C = 10$ $\gamma = 0$	0.777 ± 0.009
	MTCov	$C = 10$ $\gamma = 0.4$	**0.737 ± 0.008**	$C = 6$ $\gamma = 0.5$	**0.766 ± 0.012**	$C = 9$ $\gamma = 0.6$	**0.834 ± 0.010**

Table 3.6: Setting tpe = 0.01. Performance of methods *MT* , restricted *MTCov* and *MTCov* on three social networks and in a sampling bias framework, where the number of edges in the training set is a bit smaller than the number in the test set. RP is the random probability baseline for the accuracy of the category of caste attribute prediction. MRF indicates the maximum relative frequency linked to the covariate caste in the training set. The results are averaged over ten independent runs and the best outcomes are bolded.

			TEN 2013		TEN 2017		ALA 2017
ACCURACY for the covariate caste on the test sets	RP		0.1		0.11		0.08
	MRF		**0.311 ± 0.012**		0.321 ± 0.017		0.562 ± 0.017
	MT	$C = 12$	—	$C = 6$	—	$C = 10$	—
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.303 ± 0.033	$C = 6$ $\gamma = 0$	**0.332 ± 0.044**	$C = 10$ $\gamma = 0$	0.545 ± 0.032
	MTCov	$C = 10$ $\gamma = 0.4$	0.274 ± 0.045	$C = 6$ $\gamma = 0.5$	0.279 ± 0.045	$C = 9$ $\gamma = 0.6$	**0.571 ± 0.044**
AUC for the link prediction on the test sets	MT	$C = 12$	0.589 ± 0.007	$C = 6$	0.669 ± 0.014	$C = 10$	0.645 ± 0.008
	Restricted MTCov	$C = 12$ $\gamma = 0$	0.592 ± 0.017	$C = 6$ $\gamma = 0$	0.656 ± 0.012	$C = 10$ $\gamma = 0$	0.675 ± 0.012
	MTCov	$C = 10$ $\gamma = 0.4$	**0.637 ± 0.014**	$C = 6$ $\gamma = 0.5$	**0.687 ± 0.016**	$C = 9$ $\gamma = 0.6$	**0.759 ± 0.009**

Table 3.7: Setting tpe = 0.015. Performance of methods *MT* , restricted *MTCov* and *MTCov* on three social networks and in a sampling bias framework, where the number of edges in the test set is much higher than the number in the training set. RP is the random probability baseline for the accuracy of the category of caste attribute prediction. MRF indicates the maximum relative frequency linked to the covariate caste in the training set. The results are averaged over ten independent runs and the best outcomes are bolded.

Overall, *MTCov* achieves the highest AUC and the performances of our model in the complex settings are significantly better with respect to the comparison models. This confirms how the presence of another source of information helps in situations that lack positive examples, which in principle would guide the predictions of the test set to the over-represented 'population' in the training set. Instead, *MTCov* performed reasonably well without falling below 63%, value well above the baseline of 50% given by the purely random choice. Furthermore, our model achieves better accuracy in the majority of cases, and when *MTCov* is not the best, it is outperformed by its restriction version. Nonetheless, in the single case where our method poorly performs in the caste prediction (Table 3.7), the loss in the accuracy measure is offset by a good performance in the link prediction.

3.3.2 Interpretability

Until now, we presented a quantitative analysis, which lead to prove the robustness of our model and the importance of including the compositional dimensional when detecting communities. Now, we present a qualitative interpretation of the groups detected by *MTCov* and *MT* . Due to the lack of ground truth and to the presence of overlapping communities, there are not objective rules for analyzing or assigning labels to the communities. However, since the scaling parameter has been estimated between 0.4 and 0.6, it highlights a correlation between communi-

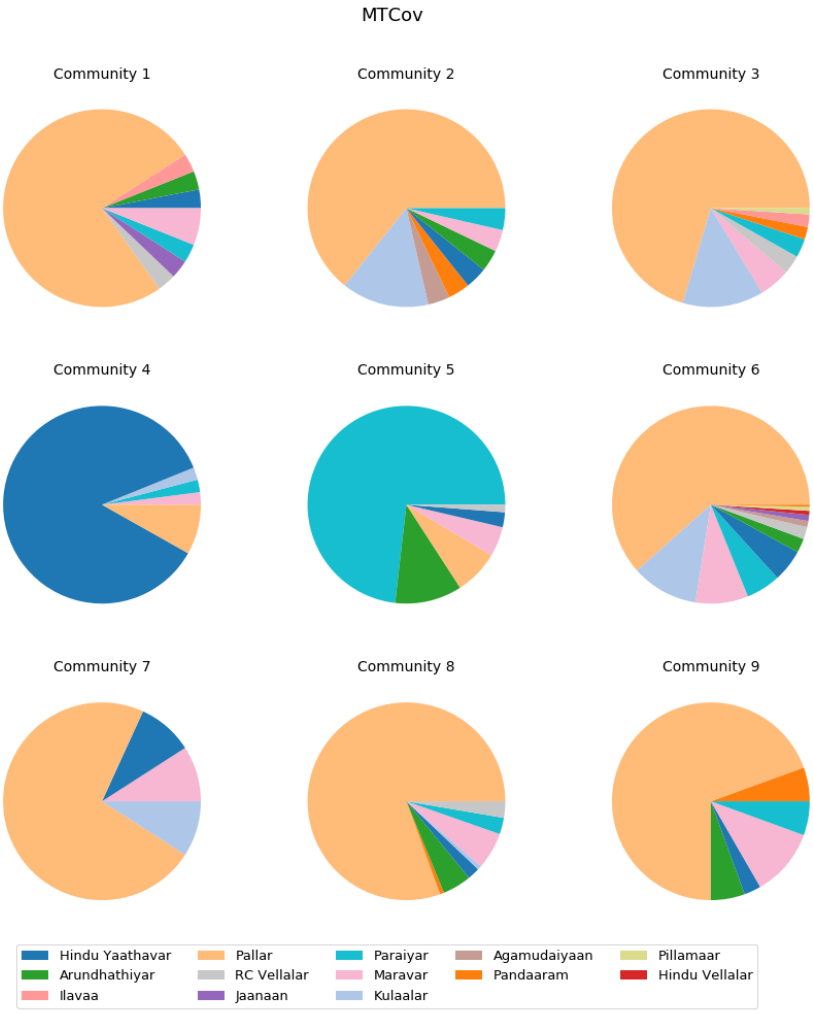


Figure 3.6: Partition of the attribute caste inside each community detected by *MTCov* algorithm in ALA 2017 network.

ties and attribute caste, which can be used as a guide for giving an interpretation of the detected groups. We investigate also if the extra groups detected by *MT* are reallocated by *MTCov* or if there is an overall topological change between the two methods. Figures 3.6 and 3.7 refer to the network ALA 2017, while the plots for the 'Tenpaṭṭi' networks are attached in Appendix B. They show the partition of the attribute caste inside each community, detected by *MTCov* and *MT* respectively. They display a notable difference in the composition of the groups with respect to the covariate: while *MT* keeps the same proportions of the categories inside each group, *MTCov* discriminates better, especially between the most representative categories, that have the maximum relative frequency in at least one community.

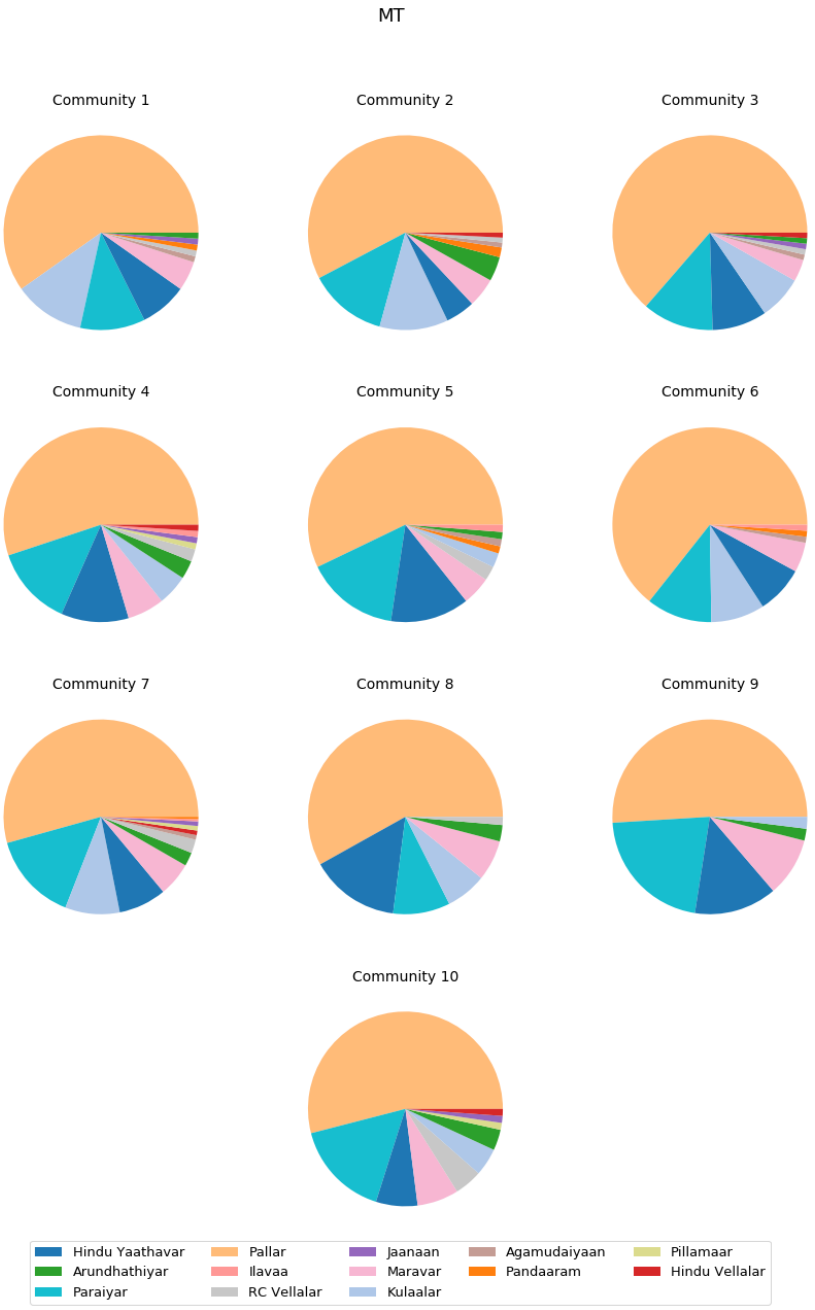


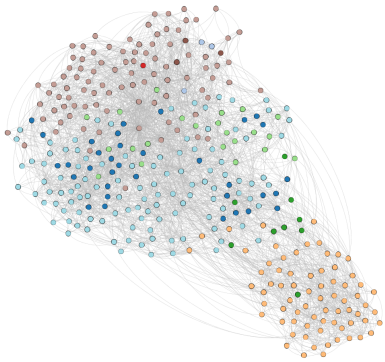
Figure 3.7: Partition of the attribute caste inside each community detected by *MT* algorithm in ALA 2017 network.

This is also confirmed by the number of communities in which each category is present, shown in Table 3.8. Although *MTCov* favors a smaller number of communities than *MT*, the differences between the two models are considerable. A clear

	TEN 2013		TEN 2017		ALA 2017	
	MT ($C = 12$)	**MTCov** ($C = 10$)	**MT** ($C = 6$)	**MTCov** ($C = 6$)	**MT** ($C = 10$)	**MTCov** ($C = 9$)
Aasaari	12	9	6	5	6	2
Agamudaiyaan	12	10	6	6	-	-
Arundhathiyar	7	7	6	5	9	6
Hindu Vellalar	1	3	3	1	5	1
Hindu Yaathavar	12	10	6	6	10	8
Ilavaa	-	-	-	-	4	2
Jaanaan	-	-	-	-	5	2
Kallar	11	9	5	2	-	-
Kulaalar	9	2	6	3	10	6
Maravar	-	-	-	-	10	9
Naayakkar	5	3	-	-	-	-
Pallar	12	10	6	6	10	9
Pandaaram	-	-	-	-	5	5
Paraiyar	-	-	-	-	10	8
Pillamaar	-	-	-	-	3	2
RC Vellalar	-	-	-	-	8	5
RC Yaathavar	12	10	6	6	-	-

Table 3.8: Number of communities in which each category is present, for each network and method.

example is given by the network TEN 2017, where a fair comparison is possible due to the same estimate of the hyperparameter C. In this framework, *MTCov* never allocates the categories in a greater number of communities than *MT*, and it keeps each of them in as few communities as possible. For this reason, the groups detected by our model seem to respect both influence of the network topology and the node attribute, creating communities that are not only structurally close but also share common characteristics. On the other hand, instead, the compositional structure of the *MT* communities is quite random and doesn't show any correlation between the communities and the covariate. However, this is at the expense of the size of the groups, which in the *MTCov* case are unbalanced. One might argue that, given that *MTCov* incorporates the covariate in the model, it should be expected to observe a correlation with the inferred communities. However, the estimate for the hyperparameter γ has been done in a range which involved also 0 (the closest situation to the *MT* case), but in all three cases the final estimate is different from the null value, underlying an interdependence between covariate, communities and structure. On the other hand, since we have never estimated $\gamma = 1$, we did not expect that the communities would fit completely the caste resulting in a clustering problem. A more complete and explanatory representation can be found in Figure 3.8. It represents the TEN 2017 network with the



(a) Caste partition.

(b) Community 1, Community 2, Community 3.

(c) Community 4, Community 5, Community 6.

Figure 3.8: TEN 2017 community partition. On top, the division by caste membership. Subplots (b) and (c) show the membership in each of the 6 communities for each node, with color ranging from white if the out-going membership $u_{ik} = 0$ to black if $u_{ik} = 1$. Values in between denote overlapping membership (grey).

grouped links of the layers, together with its predicted community assignment and the division of individuals into castes (image (a)). Subplots (b) and (c) display the membership in each of the 6 communities for each node, with a grey scale color. The black corresponds to values 1 in the membership matrix $\hat{U}$, while the white is linked with 0. The greys denote the overlapping membership. Overall, the predicted communities show a relationship with the caste partition, more evident in communities 2 and 4, and weak in community 1. Notice also the different sizes of the groups. Finally, these results suggest that there may be other attributes related to the communities.

4. Conclusion

In this thesis we present *MultiTensorCov*, a generative algorithm for community detection in multilayer networks with node attributes. It looks for overlapping communities shared between the layers, and allows different community-relationships among them, such as assortative, disassortative and core-periphery. Moreover, it can be applied both to directed and undirected networks.
Following the formalism of maximum log-likelihood estimation, this approach combines the structural and the node information into a unified log-likelihood function and provides an EM algorithm for estimating the parameters. *MTCov* combines the two sources of information through a scaling parameter γ, which measures the dependence between communities, multilayer structure and attribute.
The starting point of this thesis is the *MultiTensor* model, recently developed by De Bacco et al. (2017). Our approach extends and generalizes this work by including the compositional part given by the node features. Due to the lack of probabilistic models for community detection in annotated multilayer networks, we compare our algorithm with methods for multilayer community detection only based on the structural dimension. Thus, we used MT and the restricted version of *MTCov* with $\gamma = 0$ for validating our model.

We analyze three real social networks, which describe the social support flows between residents of two South Indian villages. Overall, *MTCov* exhibits improved performance with respect to the benchmarks both in terms of accuracy for the caste category predictions and AUC for the link predictions. Moreover, the estimated scaling parameters relies between 0.4 and 0.6, underlying a dependence between the communities and both sources of information. We show that including the compositional part, helps in providing more robust results in complex scenarios, where the positive examples in the training set have a small probability to be selected. Also, including an attribute supports the interpretation of the detected communities which are both structurally close but also share some common characteristics.

For time reasons we did not test our model against the single layer state-of-the-art, and this would be an additional step for providing robust results.

In addition, the model considers only one categorical covariate so far. A future direction is to extend *MTCov* by including more heterogeneous features. In this way, we could also exploit this formalism in terms of variables selection techniques as, ideally, the algorithm would be capable of quantifying the importance of each of them in the underlying process of community detection. Similarly, one can use link prediction for measuring the interdependence between layers and discard the redundant ones while keeping those providing independent information.

Other directions for future works include the application to other networks from different fields, which requires an automatization of the normalization procedure, as well as for the hyperparameter tuning that in this work was in part subjective and tedious. A further idea is to devise synthetic multilayer probabilistic benchmarks which incorporate also node information.

A. Appendix

A Additional Figures for Subsection 3.2.3

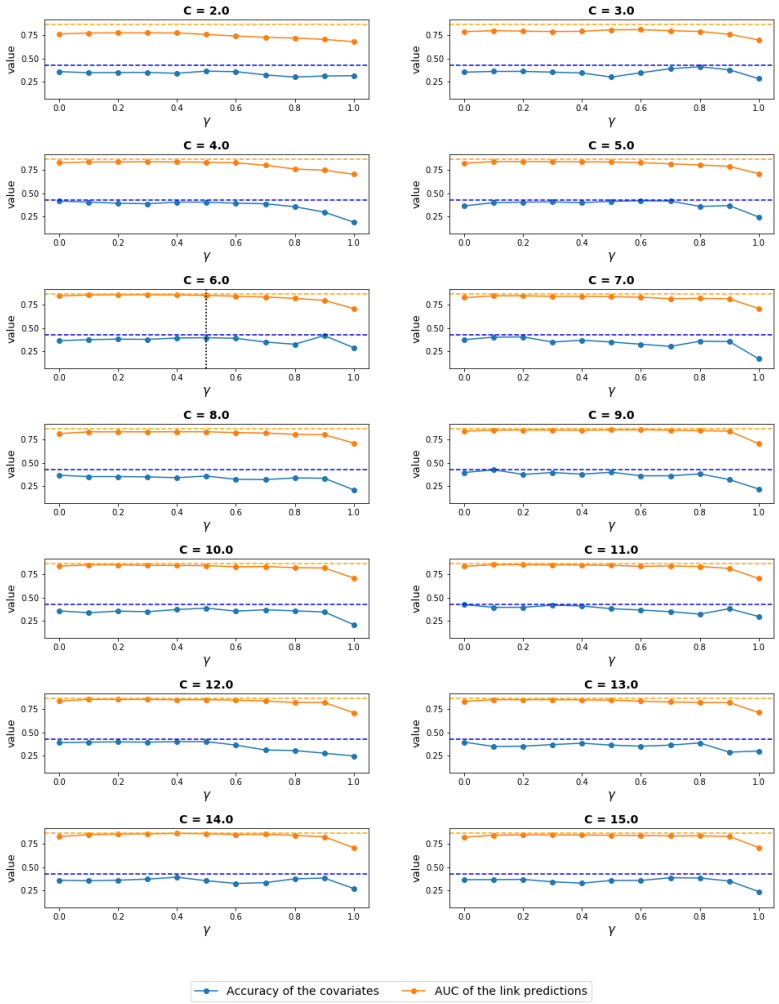


Figure A.1: 5-fold CV average values for AUC and accuracy for all combinations of the hyperparameters for the network based on Tenpaṭṭi village in the year 2017 and the attribute caste. The blue and orange dashed lines represent the best cross-validation values for the two evaluation measures independently, while the black one displays the pair of selected hyperparameters.

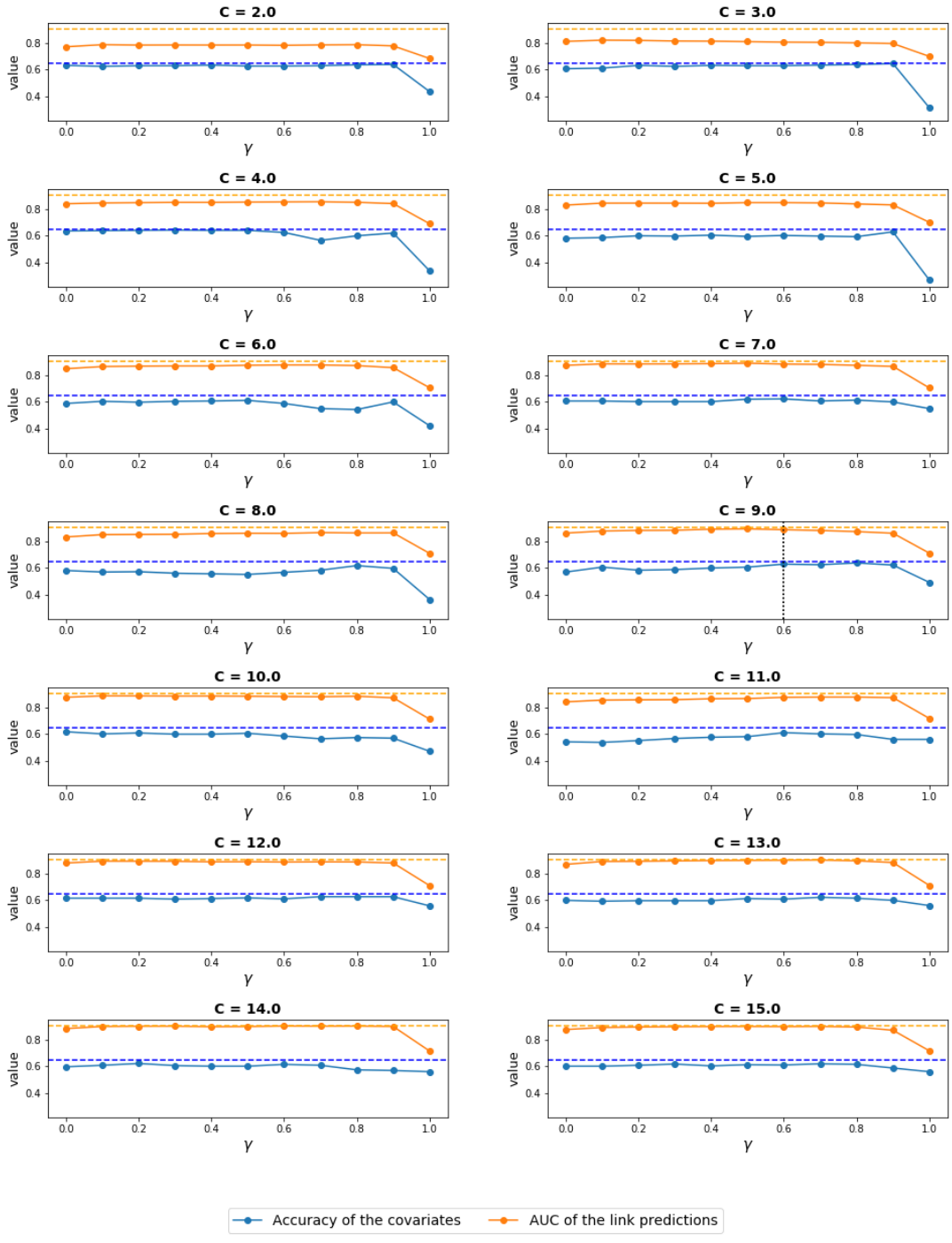


Figure A.2: 5-fold CV average values for AUC and accuracy for all combinations of the hyperparameters for the network based on Aḷakāpuram village in the year 2017 and the attribute caste. The blue and orange dashed lines represent the best cross-validation values for the two evaluation measures independently, while the black one displays the pair of selected hyperparameters.

B Additional Figures for Subsection 3.3.2

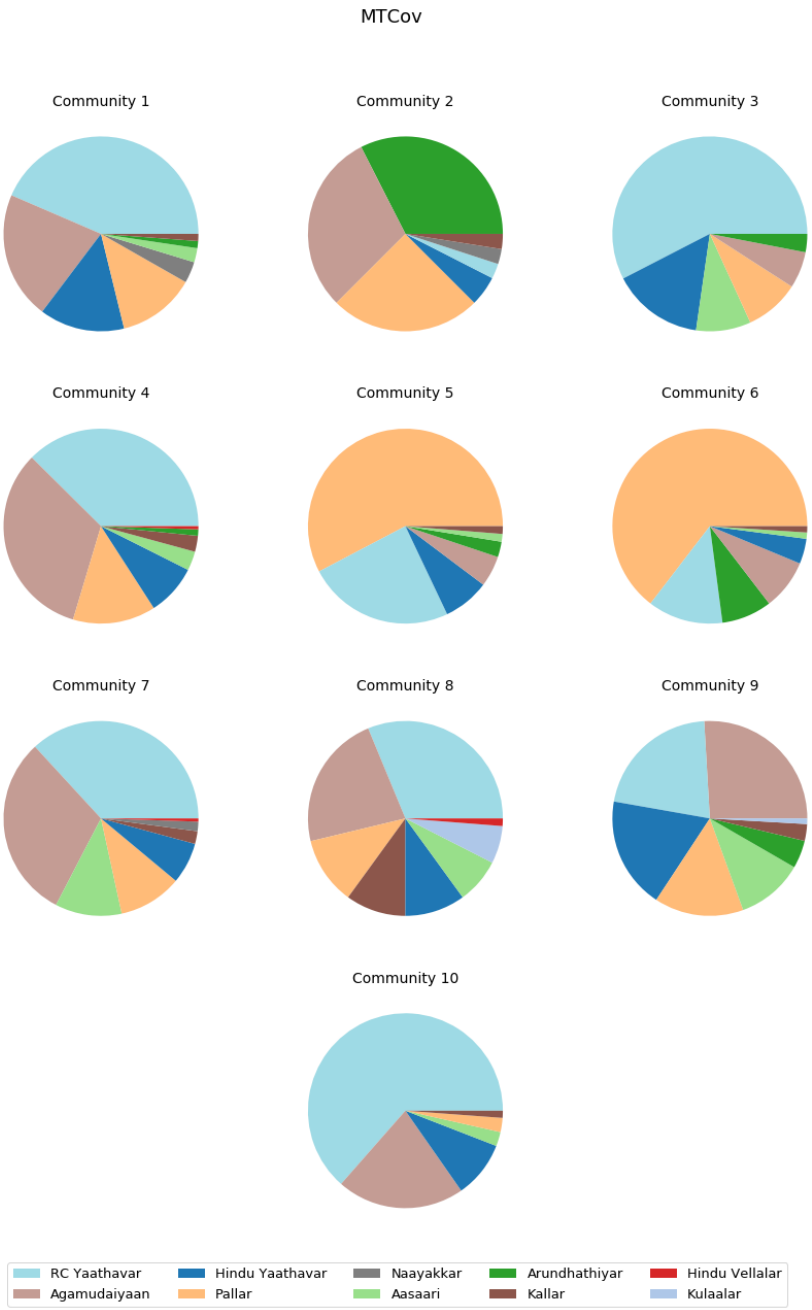


Figure A.3: Partition of the attribute caste inside each community detected by *MTCov* algorithm in TEN 2013 network.

Figure A.4: Partition of the attribute caste inside each community detected by *MT* algorithm in TEN 2013 network.

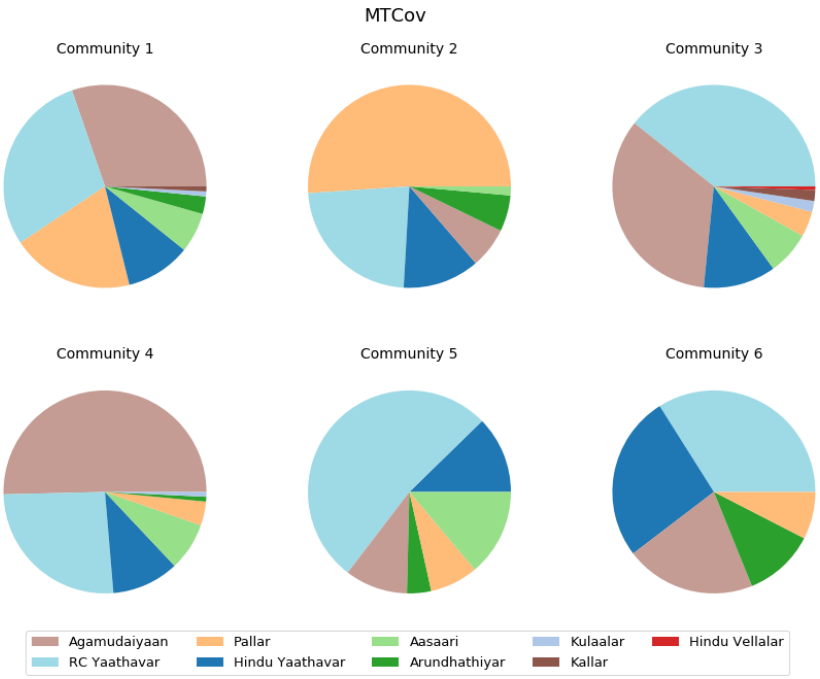


Figure A.5: Partition of the attribute caste inside each community detected by *MTCov* algorithm in TEN 2017 network.

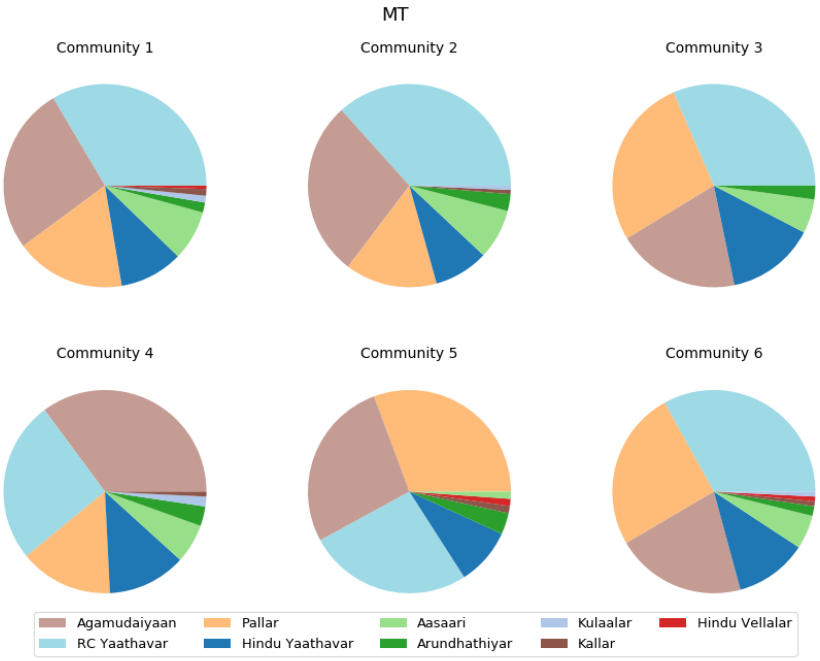


Figure A.6: Partition of the attribute caste inside each community detected by *MT* algorithm in TEN 2017 network.

B. References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2009). Mixed membership stochastic blockmodels. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems 21* (pp. 33–40). Curran Associates, Inc. Retrieved from `http://papers.nips.cc/paper/3578-mixed-membership-stochastic-blockmodels.pdf`

Barabási, A.-L., & Pósfai, M. (2016). *Network science*. Cambridge: Cambridge University Press. Retrieved from `http://barabasi.com/networksciencebook/`

Bedi, P., & Sharma, C. (2016, 02). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*, n/a-n/a. doi: 10.1002/widm.1178

Binkiewicz, N., Vogelstein, J. T., & Rohe, K. (2017). Covariate-assisted spectral clustering. *Biometrika*, *104*(2), 361–377.

Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C. I., Gómez-Gardenes, J., Romance, M., ... Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, *544*(1), 1–122.

Bothorel, C., Cruz, J. D., Magnani, M., & Micenkova, B. (2015). Clustering attributed graphs: models, measures and methods. *Network Science*, *3*(3), 408–444.

Chakraborty, T., Dalmia, A., Mukherjee, A., & Ganguly, N. (2017). Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)*, *50*(4), 54.

Chen, P.-Y., & Hero, A. O. (2017). Multilayer spectral graph clustering via convex layer aggregation: Theory and algorithms. *IEEE Transactions on Signal and Information Processing over Networks*, *3*(3), 553–567.

Clauset, A., Moore, C., & Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, *453*(7191), 98.

Coscia, M., Giannotti, F., & Pedreschi, D. (2011). A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *4*(5), 512–546.

Council, N. R. (2005). *Network science*. Washington, DC: The National Academies Press. Retrieved from `https://www.nap.edu/catalog/11516/network-science` doi: 10.17226/11516

De Bacco, C., Power, E. A., Larremore, D. B., & Moore, C. (2017, Apr). Community detection, link prediction, and layer interdependence in multilayer networks. *Phys. Rev. E*, *95*, 042317. Retrieved from `https://link.aps.org/doi/10.1103/PhysRevE.95.042317` doi: 10.1103/PhysRevE.95.042317

Dong, X., Frossard, P., Vandergheynst, P., & Nefedov, N. (2013). Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on signal processing*, *62*(4), 905–918.

Falih, I., Grozavu, N., Kanawati, R., & Bennani, Y. (2018). Community detection in attributed network. In *Companion proceedings of the the web conference 2018* (pp. 1299–1306).

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, *486*(3-5), 75–174.

Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics reports*, *659*, 1–44.

Gheche, M. E., Chierchia, G., & Frossard, P. (2018). Orthonet: Multilayer network data clustering..

Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, *99*(12), 7821–7826.

Haggerty, L. S., Jachiet, P.-A., Hanage, W. P., Fitzpatrick, D. A., Lopez, P., O'Connell, M. J., ... McInerney, J. O. (2013). A pluralistic account of homology: adapting the models to the data. *Molecular biology and evolution*, *31*(3), 501–516.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, *143*(1), 29–36.

He, D., Feng, Z., Jin, D., Wang, X., & Zhang, W. (2017). Joint identification of network communities and semantics via integrative modeling of network topologies and node contents. In *Thirty-first aaai conference on artificial intelligence.*

Hoffmann, F., Hand, D. J., Adams, N. M., Fisher, D., Guimaraes, G., et al. (2001). *Advances in intelligent data analysis: 4th international conference, ida 2001, cascais, portugal, september 13-15, 2001. proceedings* (Vol. 4). Springer Science & Business Media.

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

Hric, D., Peixoto, T. P., & Fortunato, S. (2016). Network structure, metadata, and the prediction of missing nodes and annotations. *Physical Review X*, *6*(3), 031038.

Kernighan, B. W., & Lin, S. (1970, Feb). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, *49*(2), 291-307. doi: 10.1002/j.1538-7305.1970.tb01770.x

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., & Porter, M. A. (2014). Multilayer networks. *Journal of complex networks*, *2*(3), 203–271.

Lancichinetti, A., & Fortunato, S. (2011). Limits of modularity maximization in community detection. *Physical review E*, *84*(6), 066122.

Leskovec, J., Lang, K. J., & Mahoney, M. (2010). Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on world wide web* (pp. 631–640).

Liu, W., Suzumura, T., Ji, H., & Hu, G. (2018). Finding overlapping communities in multilayer networks. *PloS one*, *13*(4), e0188747.

Malliaros, F. D., & Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. *Physics Reports*, *533*(4), 95–142.

Mercado, P., Gautier, A., Tudisco, F., & Hein, M. (2018). The power mean laplacian for multilayer graph clustering. *arXiv preprint arXiv:1803.00491*.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., & Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, *328*(5980), 876–878.

Newman, M. E., & Clauset, A. (2016). Structure and inference in annotated networks. *Nature communications*, *7*, 11863.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, *69*(2), 026113.

Peel, L., Larremore, D. B., & Clauset, A. (2017). The ground truth about metadata and community detection in networks. *Science advances*, *3*(5), e1602548.

Power, E. A. (2015). Building bigness: Religious practice and social support in rural south india. *Doctoral Dissertation. Stanford University, Stanford, CA*.

Power, E. A. (2017, 03 06). Social support networks and religiosity in rural south india. *Nature Human Behaviour*, *1*, 0057 EP -. Retrieved from `https://doi.org/10.1038/s41562-017-0057`

Rosvall, M., Delvenne, J.-C., Schaub, M. T., & Lambiotte, R. (2017). Different approaches to community detection. *arXiv preprint arXiv:1712.06468*.

Schaeffer, S. E. (2007). Graph clustering. *Computer science review*, *1*(1), 27–64.

Schein, A., Paisley, J., Blei, D. M., & Wallach, H. (2015). Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1045–1054).

Tang, F., & Ding, W. (2019). Community detection with structural and attribute similarities. *Journal of Statistical Computation and Simulation*, *89*(4), 668–685.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, *17*(4), 395–416.

Wang, Z., Hu, Y., Xiao, W., & Ge, B. (2013). Overlapping community detection using a generative model for networks. *Physica A: Statistical Mechanics and its Applications*, *392*(20), 5218–5230.

Wu, Y., Yu, H., Zhang, J., Liu, S., Huang, R., & Li, P. (2018). Usi-auc: An evaluation criterion of community detection based on a novel link-prediction method. *Intelligent Data Analysis*, *22*(2), 439–462.

Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, *45*(4), 43.

Xu, Z., Ke, Y., Wang, Y., Cheng, H., & Cheng, J. (2012). A model-based approach to attributed graph clustering. In *Proceedings of the 2012 acm sigmod international conference on management of data* (pp. 505–516).

Yang, J., McAuley, J., & Leskovec, J. (2013). Community detection in networks with node attributes. In *2013 ieee 13th international conference on data mining* (pp. 1151–1156).

Zhang, Y., Levina, E., Zhu, J., et al. (2016). Community detection in networks with node features. *Electronic Journal of Statistics*, *10*(2), 3153–3178.

Zhu, Y., Yan, X., Getoor, L., & Moore, C. (2013). Scalable text and link analysis with mixed-topic link models. In *Proceedings of the 19th acm sigkdd international conference on knowledge discovery and data mining* (pp. 473–481).