www.nature.com/scientificreports

SCIENTIFIC REPORTS
natureresearch

Check for updates

OPEN

Community detection with node attributes in multilayer networks

Martina Contisciani[1], Eleanor A. Power[2] & Caterina De Bacco[1]✉

Community detection in networks is commonly performed using information about interactions between nodes. Recent advances have been made to incorporate multiple types of interactions, thus generalizing standard methods to multilayer networks. Often, though, one can access additional information regarding individual nodes, attributes, or covariates. A relevant question is thus how to properly incorporate this extra information in such frameworks. Here we develop a method that incorporates both the topology of interactions and node attributes to extract communities in multilayer networks. We propose a principled probabilistic method that does not assume any a priori correlation structure between attributes and communities but rather infers this from data. This leads to an efficient algorithmic implementation that exploits the sparsity of the dataset and can be used to perform several inference tasks; we provide an open-source implementation of the code online. We demonstrate our method on both synthetic and real-world data and compare performance with methods that do not use any attribute information. We find that including node information helps in predicting missing links or attributes. It also leads to more interpretable community structures and allows the quantification of the impact of the node attributes given in input.

Community detection is a fundamental task when investigating network data. Its goal is to cluster nodes into communities and thus find large-scale patterns hidden behind interactions between many individual elements.

The range of application of this problem spans several disciplines. For instance, community detection has been used in sociology to analyze terrorist groups in online social networks[1]; in finance to detect fraud events in telecommunication networks[2]; in engineering to refactor software packages in complex software networks[3]; and in biology to investigate lung cancer[4] and to explore epidemic spreading processes[5]. In recent years, the variety of fields interested in this topic has broadened and the availability of rich datasets is increasing accordingly. However, most research approaches use only the information about interactions among nodes, in other words the network topology structure. This information can be complex and rich, as is the case for multilayer networks where one observes different types of interactions. For instance, in social networks, interactions could entail exchanging goods, socializing, giving advice, or requesting assistance. Most network datasets, however, contain additional information about individuals, attributes which describe their features, for instance their religion, age, or ethnicity. Node attributes are often neglected *a priori* by state-of-the-art community detection methods, in particular for multilayer networks. They are instead commonly used *a posteriori*, acting as candidates for "ground-truth" for real-world networks to measure the quality of the inferred partition[6,7], a practice that can also lead to incorrect scientific conclusions[8]. It is thus a fundamental question how to incorporate node attributes into community detection in a principled way. This is a challenging task because one has to combine two types of information[9], while evaluating the extent to which topological and attribute information contribute to the network's partition[10].

To tackle these questions, we develop MTCOV, a mathematically rigorous and flexible model to address this problem for the general case of multilayer networks, i.e., in the presence of different types of interactions. The novelty of this model relies on a principled combination of the multilayer structure together with node information to perform community detection. To the best of our knowledge, MTCOV is the first overlapping community detection method proposed for multilayer networks with node attributes. The model leverages two sources of information, the topological network structure and node covariates (or attributes), to partition nodes into communities. It is flexible as it can be applied to a variety of network datasets, whether directed, weighted, or multilayer, and it outputs overlapping communities, i.e., nodes can belong to multiple groups simultaneously. In addition, the model does not assume any *a priori* correlation structure between the attributes and the

[1]Max Planck Institute for Intelligent Systems, Cyber Valley, 72076 Tübingen, Germany. [2]Department of Methodology, London School of Economics and Political Science, London WC2A 2AE, UK. ✉email: caterina.debacco@tuebingen.mpg.de

communities. On the contrary, the contribution of the attribute information is quantitatively given as an output of the algorithm by fitting the observed data. The magnitude of this contribution can vary based on the dataset. Even if this is not very high (for instance if the attributes are noisy or sparse) the model is nevertheless able to use this extra information to improve performance. At the same time, if incorporating attribute information hurts inference tasks, the model will downweigh this contribution and instead use mostly the topological network structure.

Our method allows domain experts to investigate particular attributes and select relevant community partitions based on what type of node information they are interested in studying. In fact, by choosing the input data, we can drive the algorithm to select for communities that are more relevant to the attribute under study. If the attribute hurts performance and is consequently downweighted by the algorithm, this can be used as a signal that the attribute might not correlate well with any partition, given the remaining topological information available, and thus inform the expert accordingly.

We study MTCOV on synthetic multilayer networks, a variety of single-layer node-attributed real networks and several real multilayer networks of social support interactions in two Indian villages. We measure performance based on prediction tasks and overlap with ground-truth (when this is known). For single-layer networks, we compare the performance of MTCOV to state-of-the-art community detection algorithms with node attributes; for multilayer networks, we test against a state-of-the-art algorithm that does not use any node attribute information and measure the extent to which knowing both types of information helps inference. We find that MTCOV performs well in predicting missing links and attributes. It also leads to more interpretable community structures and allows the quantification of the impact of the node attributes given as input.

To summarize, we present MTCOV, a new method that incorporates both the topology of interactions and node attributes to extract communities in multilayer networks. It is flexible, efficient and it has the property of quantitatively estimating the contributions of the two sources of information. It helps domain experts to investigate particular attributes and to better interpret the resulting communities. Moreover, by including relevant node attributes, it boosts performance in terms of edge prediction.

Related work.

Several methods have been proposed to study community detection in networks[11]. In particular, we are interested in those valid for multilayer networks[12]. These generalize single-layer networks in that they can model different types of interactions and thus incorporate extra information that is increasingly available. Among these, we focus on generative models for multilayer networks[13–19], which are based on probabilistic modeling like Bayesian inference or maximum likelihood optimization. These are flexible and powerful in terms of allowing multiple inference tasks, injecting domain knowledge into the theoretical framework, and being computationally efficient. However, the majority of these methods do not consider node attributes as input along with the network information. In fact, the few methods developed for community detection in multilayer networks with node attributes are based on first aggregating the multilayer network into a single layer, either by combining directly the adjacency matrices of each layer[20] or by using similarity matrices derived from them along with the node attributes[21,22]. In the context of data mining, a similar problem can be framed for learning low dimensional representations of heterogeneous data with both content and linkage structure (what we call attributes and edges). This is tackled by using embeddings extracted via deep architectures[23], which is rather different than our approach based on statistical inference. Our problem bears some common ground with the one studied by Sachan et al.[24] for extracting communities in online social networks, where users gather based on common interests; they adopt a Bayesian approach, but with a rather different goal of associating different types of edges to topics of interest. A related but different problem is that of performing community detection with node attributes on multiple independent networks[25,26]; this differs with modeling a single multilayer network in that it assumes that covariates influence in the same way all the nodes in a network but in a different way the various networks in the ensemble. For single-layer networks, there has been more extensive work recently on incorporating extra information on nodes[9,25,27–34]. Among those adopting probabilistic modeling, some incorporate covariate information into the prior information of the latent membership parameters[25,35,36], while others include covariates in an additive way along with the latent parameters[37,38], so that covariates influences the probability of interactions independently of the latent membership.

These works show the impact of adding nodes attributes in community detection *a priori* into the models to uncover meaningful patterns. One might then be tempted to adopt such methods also in multilayer networks by collapsing the topological structure into a suitable single network that can then be given in input to these single-layer and node-attributed methods as done by Gheche et al.[20]. However, collapsing a multilayer network often leads to important loss of information, and one needs to be careful in determining when this collapse is appropriate and how it should be implemented, as shown for community detection methods without attribute information[39,40]. Thus the need of a method that not only incorporates different types of edges but also node attributes.

Results

We test MTCOV's ability to detect communities in multilayer networks with node attributes by considering both synthetic and real-world datasets. We compare against MULTITENSOR[13], an algorithm similar to ours but that does not include node attributes. We also test MTCOV's performance on single-layer networks, as the mathematical framework behind MTCOV still applies. Given this potential use and the paucity of algorithms suitable for comparison for multilayer networks, such comparisons assess the general utility of MTCOV.

Multilayer synthetic networks with ground-truth.

To illustrate the flexibility and the robustness of our method, we generate multilayer synthetic networks with different kinds of structures in the various layers adapting the protocol described in De Bacco et al.[13] to accommodate node attributes. We generate attributes as

	G_1				G_2				G_3			
Method	**F1-score**	**Jaccard**	**CS**	**L_1**	**F1-score**	**Jaccard**	**CS**	**L_1**	**F1-score**	**Jaccard**	**CS**	**L_1**
MULTITENSOR	0.512 ± 0.006	0.344 ± 0.006	0.585 ± 0.005	0.492 ± 0.004	0.514 ± 0.006	0.346 ± 0.06	0.614 ± 0.005	0.490 ± 0.005	**0.999 ± 0.001**	**0.998 ± 0.001**	**0.991 ± 0.001**	0.063 ± 0.002
MTCOV_0.3	0.7 ± 0.2	0.5 ± 0.2	0.7 ± 0.1	0.4 ± 0.1	0.8 ± 0.2	0.7 ± 0.2	0.8 ± 0.1	0.3 ± 0.2	0.995 ± 0.002	0.990 ± 0.004	0.984 ± 0.002	0.080 ± 0.004
MTCOV_0.5	0.6 ± 0.1	0.5 ± 0.2	0.7 ± 0.1	0.4 ± 0.1	0.992 ± 0.005	0.985 ± 0.009	0.986 ± 0.004	0.064 ± 0.004	0.996 ± 0.002	0.992 ± 0.004	0.985 ± 0.002	0.079 ± 0.004
MTCOV_0.7	**0.988 ± 0.002**	**0.976 ± 0.004**	**0.977 ± 0.002**	0.079 ± 0.003	**1. ± 0.**	**1.000 ± 0.001**	**0.991 ± 0.001**	0.062 ± 0.002	0.994 ± 0.002	0.988 ± 0.004	0.982 ± 0.001	0.087 ± 0.002
MTCOV_0.9	0.958 ± 0.003	0.920 ± 0.005	**0.977 ± 0.001**	**0.050 ± 0.002**	0.992 ± 0.002	0.984 ± 0.004	0.988 ± 0.001	**0.050 ± 0.002**	0.976 ± 0.003	0.952 ± 0.006	0.982 ± 0.002	**0.051 ± 0.003**

Table 1. Performance of algorithms MULTITENSOR and MTCOV on synthetic multilayer networks with attributes. We use different matches (one per row, e.g., MTCOV_0.3 denotes a match of 0.3, this is also the value we use to fix γ) between attributes and planted communities on synthetic directed multilayer networks. Results are averages and standard deviations over 10 networks samples for each network type G_m, $m = 1, 2, 3$; we take the average performance over the incoming and outgoing memberships, i.e., the matrices U and V, and the best performances are in boldface. Networks are generated with stochastic block model with $C = 2$, $N = 1000$ and average degree $k = 4$. Denote W^a, W^d, W^{cp} and W^{bd}, the affinity matrices of the assortative, disassortative, core-periphery and the biased directed layers respectively. Then, their entries are $w^a_{11} = w^a_{22} = w^d_{12} = w^d_{21} = w^{cp}_{11} = w^{bd}_{12} = \frac{kC}{N}$, $w^a_{12} = w^a_{21} = w^d_{11} = w^d_{22} = w^{cp}_{12} = w^{cp}_{21} = w^{bd}_{11} = w^{bd}_{22} = 0.1 \times \frac{kC}{N}$ and $w^{cp}_{22} = w^{bd}_{12} = 0.03 \times \frac{kC}{N}$. The F1-score, Jaccard, CS and L_1 are performance metrics as defined in the "Methods" section.

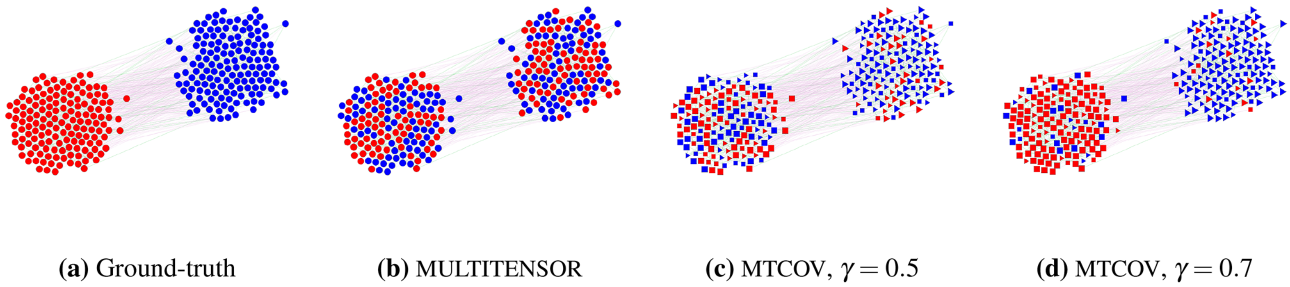
(a) Ground-truth **(b)** MULTITENSOR **(c)** MTCOV, $\gamma = 0.5$ **(d)** MTCOV, $\gamma = 0.7$

Figure 1. Partition of a synthetic multilayer network with attributes. We generated synthetic directed multilayer networks using a stochastic block model, that aligns with G_1. To illustrate, here we do the equivalent task on a smaller network of size N = 299, C = 2 communities of equal-size unmixed group membership and L = 2 layers, of which one is assortative (green) and one disassortative (pink); (**a**) the ground-truth partition; (**b**–**d**) the communities inferred by three different methods: (**b**) MULTITENSOR, an algorithm without attributes, (**c**) MTCOV using the network structure and the attributes with the same proportion, i. e. $\gamma = 0.5$ and (**d**) MTCOV using mostly the attribute structure, i.e. $\gamma = 0.7$. Colors denote the inferred partition; the attributes in (**c**) and (**d**) are generated by matching them with true community assignments for the 50% and 70% of the nodes respectively, and chosen uniformly at random from the non-matching values; square and triangle denote the synthetic dummy attribute (squares are matched with the red group, triangles with the blue) and the size of the node shows the nodes matched with the true community (bigger means deterministic match, smaller means uniform at random match). We use the matrix U for the membership.

done in Newman and Clauset[27]: we match them with planted communities in increasing ratios varying from 0.3 to 0.9; these values correspond also to the γ parameters that we fix for MTCOV. Specifically, we generate three types of directed networks using a stochastic block model[41], all with $C = 2$ communities of equal-size unmixed group membership and $N = 1000$ nodes, but with different numbers and kinds of layers, similar to De Bacco et al.[13]. The first network (G_1) has $L = 2$ layers, one assortative (W^α has higher diagonal entries) and one disassortative (W^α has higher off-diagonal entries); the second (G_2) has $L = 4$ layers, two assortative and two disassortative and the third (G_3) has $L = 4$ layers, one assortative, one disassortative, one core-periphery (W^α has higher diagonal entries but one of the two is bigger than the other) and one with biased directed structure (W^α has higher off-diagonal entries but one of the two is bigger than the other). We generate ten independent samples of each of these types of networks and use all the evaluation metrics described in the "Methods" section in the presence of ground-truth. We use the membership inferred by the algorithms using the best maximum likelihood fixed point over 10 runs with different random initial conditions. As shown in Table 1, MTCOV performs significantly better than MULTITENSOR on the first and second network. This suggests that incorporating attribute information can significantly boost inference, with an increasing benefit for a smaller number of layers. Figure 1 shows an example of this result. Notice that G_2 requires a smaller match ($\gamma = 0.5$) between attributes and communities than G_1 ($\gamma = 0.7$) to achieve similar performance. G_1 and G_2 have similar structure, but the second has twice as many layers. Thus, increasing the number of layers may require less contribution from the extra information of the attributes, a possible advantage for multilayer networks. This intuition is reinforced by noticing not only that the best performance is achieved for $\gamma < 0.9$, but also that both the algorithms perform very well in the third network, regardless of the value of the match between attributes and communities. Con-

Village	Year	Nodes	Edges	$\langle k \rangle$	Caste	Religion	Age	Gender
Aḻakāpuram	2013	419	4,161	20	14	3	11	2
	2017	441	5,578	25	13	3	12	2
Teṉpaṭṭi	2013	362	3,374	19	11	2	11	2
	2017	346	3,806	22	9	2	12	2

Table 2. Network summary statistics for the four social support networks of Indian villages. Each has the same set of 6 layers and Edges are the total over them; $\langle k \rangle$ is the average degree per node on the whole multilayer network. The columns Caste, Religion, Age and Gender are the number of different categories observed in each network for their respective attribute.

trary to G_2, G_3 has a different structure in each of the 4 layers. This diversity can be even more beneficial than having more but correlated layers (as in G_1 vs. G_2). These synthetic tests demonstrate the impact of leveraging both node attributes and topological information: when topological structure is not very informative (as in G_1 with only two layers), adding node attributes can significantly help in recovering the communities. In contrast, when topological information is more complex (as in G_3 where all layers are different), properly combining the different layers' structures can compensate for a limited access to extra information on nodes. Overall, this shows the need for methods suited for exploiting various sources of information and the complexity behind multilayer networks.

Multilayer social support network of rural Indian villages.

We demonstrate our model beyond synthetic data by applying it to social support networks of two villages in Tamil Nadu, India, which we call by the pseudonyms "Teṉpaṭṭi" (Ten) and "Aḻakāpuram" (Ala)[42–44]. Data were collected in the form of surveys where adult residents were asked to nominate those individuals who provided them with various types of support, including running errands, giving advice, and lending cash or other household items. These were collected in two rounds, one in 2013 and the other in 2017. Each type of support corresponds to a layer in the network; we consider only those layers present in both rounds, for a total of $L = 6$ layers. After pre-processing the data, by considering only those individuals who had at least one outgoing edge and removing self-loops, the resulting networks have the size reported in Table 2. In addition, several attributes were collected, which include information about age, religion, caste, and education level. Ethnographic observation in these villages[42] and previous analyses[43,44] suggest that social relations are strongly structured by religious and caste identity, with these divisions shaping where people live, who they marry, and who they choose to associate with. In other words, they suggest a dependence between the attributes Religion and Caste and the mechanisms driving edge formation in these social support networks. Motivated by these insights, here we consider the attributes Caste and Religion and add them into the model. In addition, we test the importance of variables that we expect to be less informative, such as gender and age. The latter, being continuous, is also an example of a non-categorical variable. Provided it has a finite range, as it is the case for age, we can encode it into categorical by binning its values. Here we use equal bins of size 5 years.

Without assuming *a priori* any ground-truth, we measure performance using the AUC and accuracy as explained in the "Methods" section. We compare with MULTITENSOR to measure the extent to which adding the attributes helps predicting edges and attributes; in addition, in terms of accuracy values, we consider two baselines for further comparisons: (1) a uniform at random probability over the number of possible categories (RP); and (2) the maximum relative frequency of the attribute value appearing more often (MRF). We fix hyperparameters using 5-fold cross-validation along with grid-search procedure (see "Cross-validation tests and hyperparameter settings" section for more details). We obtain values of $\gamma \in [0.2, 0.9]$, signalling relevant correlations between attributes and communities. For details, see Supplementary Table S2. Empirically, we observe that when $\gamma > 0.5$ the algorithm achieves better performance in terms of link and attribute prediction by well balancing the log-likelihood of the attribute dimension and the one of the network structure.

For validation, we split the dataset into training/test sets uniformly at random as explained in the "Methods" section. Table 3 reports the average results over ten runs for each network, and shows that MTCOV is capable of leveraging two sources of information to improve both performance metrics. In fact, our algorithm systematically achieves the highest accuracy for attribute prediction and the highest AUC for edge prediction (boldface). While a good performance in attribute prediction is expected by design as we add this data into the model, the fact that it also boosts performance in terms of edge prediction is not granted *a priori*. Instead, it is a quantitative way to show that an attribute plays an important role in the system. It also demonstrates the potential of capturing correlations between two different sources of information, which can have relevant applications, in particular when missing information of one kind. Notice in particular the improvement in AUC when using caste compared to no attribute given (MULTITENSOR). The other attributes are less informative; in particular age has a performance similar to MULTITENSOR in edge prediction, signalling that it does not contribute to inform edge formation. Indeed, it has the smallest inferred γ (always < 0.5), which gives also worse accuracy performance than the baseline, signalling again that this attribute may not be correlated with the community structure. All these results show the flexibility of MTCOV in adapting based on the data given in input: if warranted, it is able to ignore those attributes that are not correlated with network division and instead find communities that are mainly based on the network structure. Next, we test how adding node attributes impacts robustness against unbalanced data, where the ratio of positive examples (existing edges) observed in the training is different than that in the test set. We denote the total probability of selecting an edge in the test as *tpe*

Attribute	Method	ACCURACY for attribute prediction				AUC for link prediction			
		Ala 2013	Ala 2017	Ten 2013	Ten 2017	Ala 2013	Ala 2017	Ten 2013	Ten 2017
	MULTITENSOR					0.771 ± 0.009	0.835 ± 0.006	0.758 ± 0.005	0.81 ± 0.01
Caste	RP	0.07	0.08	0.10	0.11				
	MRF	0.556 ± 0.009	0.57 ± 0.01	0.32 ± 0.01	0.33 ± 0.02				
	MTCOV	**0.80 ± 0.05**	**0.77 ± 0.05**	**0.69 ± 0.09**	**0.74 ± 0.07**	**0.837 ± 0.009**	**0.858 ± 0.008**	**0.829 ± 0.006**	**0.82 ± 0.01**
Religion	RP	0.33	0.33	0.50	0.50				
	MRF	0.837 ± 0.008	0.843 ± 0.006	0.696 ± 0.008	0.679 ± 0.008				
	MTCOV	**0.96 ± 0.02**	**0.95 ± 0.03**	**0.76 ± 0.08**	**0.80 ± 0.05**	0.813 ± 0.007	0.83 ± 0.01	0.81 ± 0.02	0.80 ± 0.01
Age	RP	0.09	0.08	0.09	0.08				
	MRF	**0.135 ± 0.005**	**0.126 ± 0.007**	0.126 ± 0.005	**0.128 ± 0.008**				
	MTCOV	0.11 ± 0.03	0.11 ± 0.02	**0.13 ± 0.04**	0.10 ± 0.03	0.80 ± 0.01	0.823 ± 0.008	0.783 ± 0.009	0.80 ± 0.01
Gender	RP	0.50	0.50	0.50	0.50				
	MRF	0.584 ± 0.009	0.58 ± 0.01	0.56 ± 0.01	0.55 ± 0.01				
	MTCOV	**0.61 ± 0.05**	**0.65 ± 0.04**	**0.58 ± 0.08**	**0.71 ± 0.08**	0.79 ± 0.02	0.831 ± 0.009	0.80 ± 0.01	0.81 ± 0.01

Table 3. Prediction performance on real multilayer networks with attributes. Results are averages and standard deviations over 10 independent trials of cross-validation with 80–20 splits selected uniformly at random (i.e., *tpe* = 0.004); the best performances are in boldface. Datasets are described in Table 2. RP is the performance of uniform random probability and MRF the one of the maximum relative frequency, see "Methods" section for details.

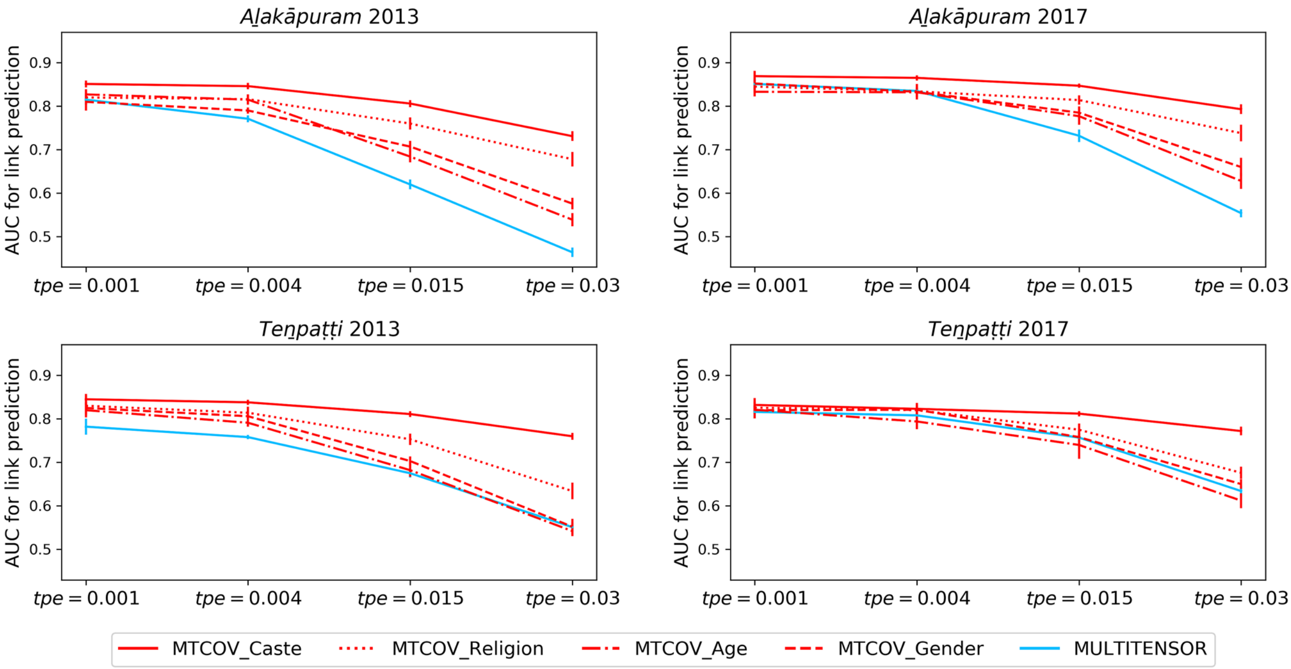

Figure 2. Probabilistic link prediction with biased edge sampling. Results are AUC values of MTCOV and MULTITENSOR on four social support networks in different held-out settings. Here *tpe* indicates the total probability of selecting one edge (positive example) in the test set. We consider Caste, Religion, Age and Gender attributes; results are averages and standard deviations over 10 independent runs.

and consider values $tpe \in \{0.001, 0.004, 0.015, 0.03\}$ denoting under-representation (0.001), equal (0.004), and over-representation (values 0.015 and 0.03) compared to the uniform at random selection (empirically we find $tpe = 0.004$). In these tests, we hold out 20% of the entries of A biasing their selection using the *tpe* values; in addition, we give as input the whole design matrix X (attributes) and measure link prediction performance. We observe that MTCOV is significantly more robust than the algorithm that does not use any attribute information, regardless of the value of γ. In fact, even though performance deteriorates as we decrease the number of positive examples in the training set (i.e., higher *tpe*), MTCOV is less impacted by this, as shown in Fig. 2 (results reported in Supplementary Table S3). Notice in particular performance discrepancies when using the attribute

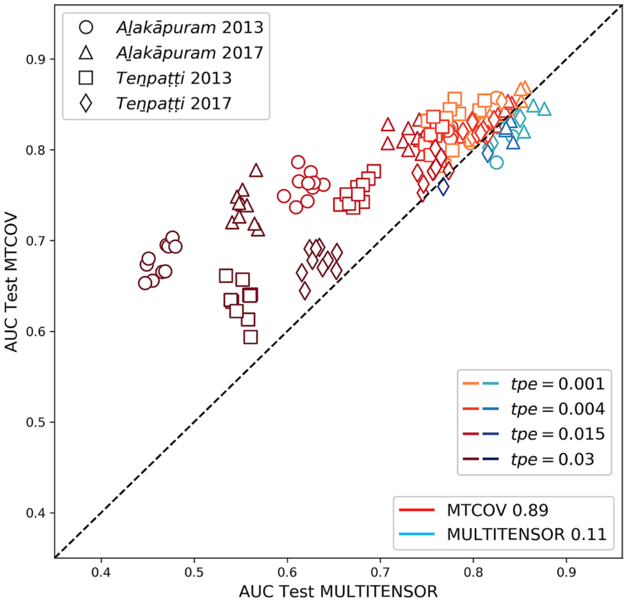

Figure 3. Trial-by-trial probabilistic link prediction with biased edge sampling. The values of AUC for MTCOV and MULTITENSOR are shown on the vertical axis and the horizontal axis respectively. The brightness represents the hardness of the settings in terms of biasing the edge sampling in the training. From bottom to top: $tpe = 0.03$ (hard, dark color), $tpe = 0.015$, $tpe = 0.004$ (random), $tpe = 0.001$ (easy, light color). Points above the diagonal, shown in shades of red, are trials where MTCOV is better performing than MULTITENSOR. The fractions for which each method is superior are shown in the plot legend. We use the attribute Religion.

Caste in the difficult regimes ($tpe \in \{0.015, 0.03\}$): MTCOV's performance deteriorates only a little, while using the other attributes or no attribute makes performance significantly worse, with AUC down to 0.6 from a value higher than 0.8 for easier regimes. Moreover, notice that attributes with the same scaling parameter value can give different prediction results, underlying the necessity to consider both the value of the estimated γ and the quality of the attribute to quantify its importance. This could explain why Caste provides always better results, given by the fact that its categories are more heterogeneous (i.e., more information) than Religion and Gender. The robustness of MTCOV is also confirmed by analyzing the performances on a trial-by-trail basis, each trial being a random sample of the held-out entries. As we show in Fig. 3, MTCOV better predicts links in 89% of the trials and never goes below the threshold of 0.5, the baseline random choice. These results demonstrate how adding another source of information helps when observing a limited amount of network edges.

Qualitative analysis of a social support network. To demonstrate our MTCOV model beyond prediction tasks and highlight its potential for interpretability, we show as an example its qualitative behavior on the real network of Aḻakāpuram in 2017 (see Table 2). Specifically, we compare the communities extracted by our algorithm and those inferred by MULTITENSOR. To ease comparison, we fix the same number of groups to $C = 4$ for both algorithms and measure how caste membership distributes across communities, and fix $\gamma = 0.8$ as obtained with cross-validation. Figure 4 shows the magnitude of each individual's inferred outgoing memberships u_i in each group. While the communities identified by MTCOV and MULTITENSOR show substantial similarities, MTCOV generally classifies castes more consistently into distinct communities, as we show in Figs. 4 and 5. To make a quantitative estimate of the different behaviors, we measure the entropy of the attribute inside each community $H_k = -\sum_{z=1}^{Z} f_z \log f_z / \log(Z)$, where f_z is the relative frequency of the z-th caste inside a group k, and the denominator is the entropy of a uniform distribution over the Z castes, our baseline for comparison. Values of H_k close to 1 denote a more uniform distribution of castes, whereas smaller values denote an unbalanced distribution with most of the people belonging to a few castes. We find that MTCOV has smaller entropies over the groups, with two groups having the smallest values, whereas MULTITENSOR has the highest, showing its tendency to cluster individuals of different castes into the same group. In addition, we observe that MTCOV has a more heterogenous group size distribution which seems to be correlated with caste. Notably, the algorithms differ in how they place two caste groups that live in hamlets separated from the main village (the Hindu Yātavars and CSI Paṟaiyars). With MULTITENSOR, they are grouped together, while with MTCOV, the Hindu Yātavars are joined up into a community with Paḷḷars and Kulālars. While MULTITENSOR is clearly picking up the structural similarities of the two hamlets, this division makes little sense socially and culturally. In contrast, the way in which MTCOV defines a community which spans caste boundaries (MTCOV C1) aligns with ethnographic knowledge of the relations between these castes. Finally, we remark that there might be multiple meaningful community divisions in the network, and the fact that MTCOV's partition seems to better capture the distributions in the attribute caste does not mean than one algorithm is better than the other. In fact, there might be other hidden topological properties that MULTITENSOR's partition is picking up by being agnostic to

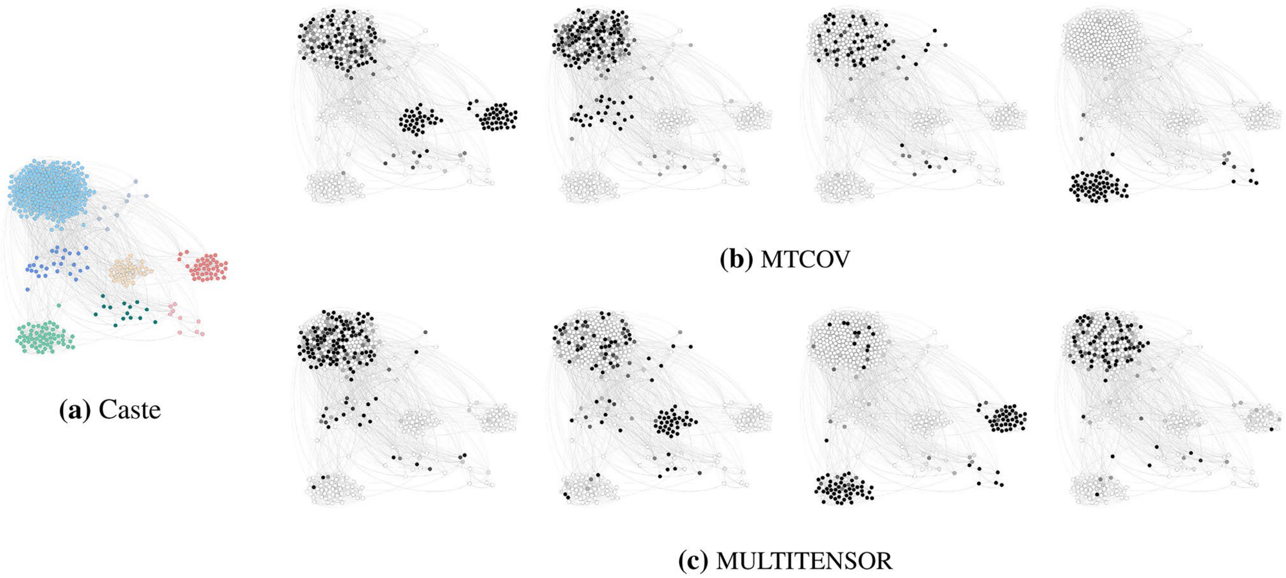

Figure 4. Attributes and inferred communities. Nodes of the social support network of Aḻakāpuram in 2017 are colored by: (**a**) the attribute Caste (with colors as shown in Fig. 5); inferred communities by (**b**) MTCOV and (**c**) MULTITENSOR. Darker values in the grey scales indicate higher values of the entry of the membership vector u_i.

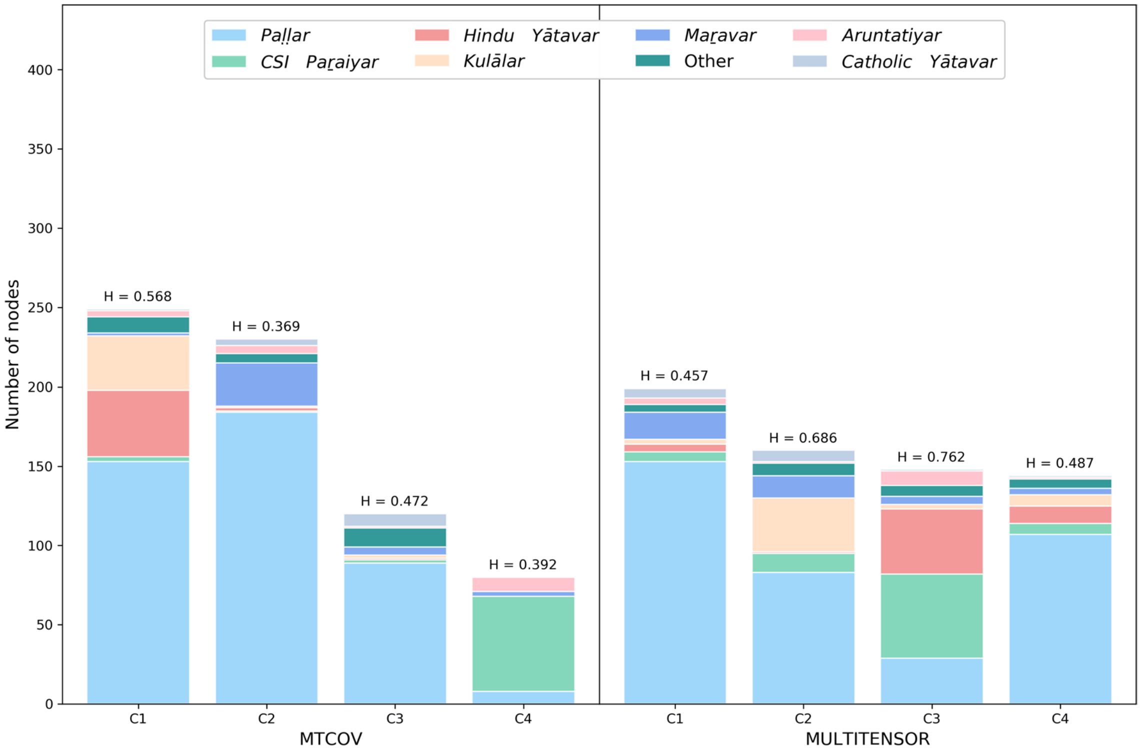

Figure 5. Partition of the attribute Caste inside each community detected by MTCOV and MULTITENSOR in the social support network of Aḻakāpuram in 2017. The category Other contains small categories having less than five individuals. The label on top of each bar is the value of the entropy of the variable Caste inside the corresponding community. Note that nodes can have mixed membership, here we build a group k by adding to it all nodes i that have a non-zero k-th entry u_{ik}. The number of nodes is $N = 441$, corresponding to the maximum value of the y-axis plotted.

Method	F1-score			Jaccard similarity		
	facebook	football	polblogs	facebook	football	polblogs
MTCOV	**0.5 ± 0.1**	**0.86 ± 0.03**	0.8 ± 0.2	**0.4 ± 0.1**	**0.82 ± 0.04**	0.8 ± 0.2
NC	0.48 ± 0.08	0.82 ± 0.06	**0.95 ± 0.09**	0.36 ± 0.08	0.75 ± 0.08	**0.9 ± 0.1**
CESNA	0.46 ± 0.09	0.7 ± 0.0	0.6 ± 0.0	0.33 ± 0.08	0.6 ± 0.0	0.4 ± 0.0

Table 4. Performance of methods MTCOV, NC and CESNA on three datasets, according to two different measures used in the Eq. (16). The results are averages and standard deviations over ten independent runs and the best outcomes are bolded.

caste membership. The choice of which algorithm to use should be made based on the final goal of the application at hand.

Results on single-layer networks. Our model can be used for single-layer networks as well. For these we can compare against two state-of-the-art algorithms, both probabilistic generative models but different in their assumptions: CESNA[9] which considers overlapping communities and posits two independent Bernoulli distributions for network edges and node attributes; and the model proposed by Newman and Clauset[27] (NC) for non-overlapping communities, a Bayesian approach where the priors on the community memberships depend on the node attributes. CESNA, similarly to our model, assumes conditional independence of the two likelihoods and introduces a regularization parameter between them; it uses block-coordinate ascent for parameters' estimation, while NC uses an EM algorithm for parameters' estimation, similarly to what we do here. We test MTCOV against them on both synthetic and real single-layer networks with node attributes, with and without ground-truth. We transform directed networks to undirected because both CESNA and NC do not distinguish for edge directionality. Results on synthetic data show that MTCOV and NC have similar performance in correctly classifying nodes in their ground-truth communities and both are better than CESNA; the main difference is that MTCOV is more stable and has less variance for high attribute correlation, in particular in the hard regime where classification is more difficult. We leave details in the Supplementary Section S4. For single-layer real networks, we use datasets with ground-truth candidates and node attributes: the ego-Facebook network (*facebook*)[45], a set of 21 networks built from connections between a person's friends where potential ground-truth are circles of friends hand-labeled by the ego herself; the American College football network (*football*)[46], a network of football teams playing against each other, where a ground-truth candidate is the conference to which each team belongs; and a network of political blogs (*polblogs*)[47] where potential ground-truth communities are divided by *left/liberal* and *right/conservative* political parties, see Supplementary Section S4 for details. For each network, we run a 5-fold cross-validation procedure combined with grid-search for fixing the hyperparameter γ (see "Cross-validation tests and hyperparameter settings" section for details; note that in this case we use the ground-truth value of C, hence γ is the only hyperparameter left to be tuned). For *facebook* we find that the average over the 21 networks is $\gamma = 0.15$, which signals a low correlation between the covariates and the communities, whereas for the *football* and *polblogs* networks we obtain much higher values of γ equal to 0.6 and 0.75 respectively. MTCOV has better performance in terms of F1-score and Jaccard similarity across the majority of datasets, as shown in Table 4. This is also supported by a trial-by-trial comparison shown in Fig. 6 for F1-score (similar results are obtained for Jaccard), where we find that MTCOV is more accurate in 59% and 90% of the cases than NC and CESNA, respectively.

Discussion

We present MTCOV, a generative model that performs overlapping community detection in multilayer networks with node attributes. We show its robustness in adapting to different scenarios, and its flexibility in exploiting the attributes that are more informative while ignoring those that are less correlated with the network communities. Our method is capable of estimating quantitatively the contribution given by the attributes and incorporating them to improve prediction performance both in terms of recovering missing attributes and in terms of link prediction. This allows domain experts to investigate particular attributes and select relevant community partitions based on what type of node information they are interested in investigating. There are valuable possible extensions of this work. One example is to incorporate modeling of more complex data types for the attributes, for instance combinations of discrete and continuous attributes, or other types of extra information, like time-varying network elements, whether the attributes, node, edges or combinations of these. From a technical point of view, when the topological and attribute datasets are very unbalanced in size, this might impact their relative likelihood weight and thus inference. One should then consider automating the process of rescaling them accordingly, as a pre-processing step to be incorporated into the model. Similarly, hyperparameter selection would benefit from an automatized routine when more than one performance metric is considered. The relations between attributes and communities could be transferred across networks to predict missing information when having access to similar but incomplete datasets. We show examples of these here, where we studied two snapshots of the same village networks across time. While we leave these questions for future work, we provide an open source version of the code.

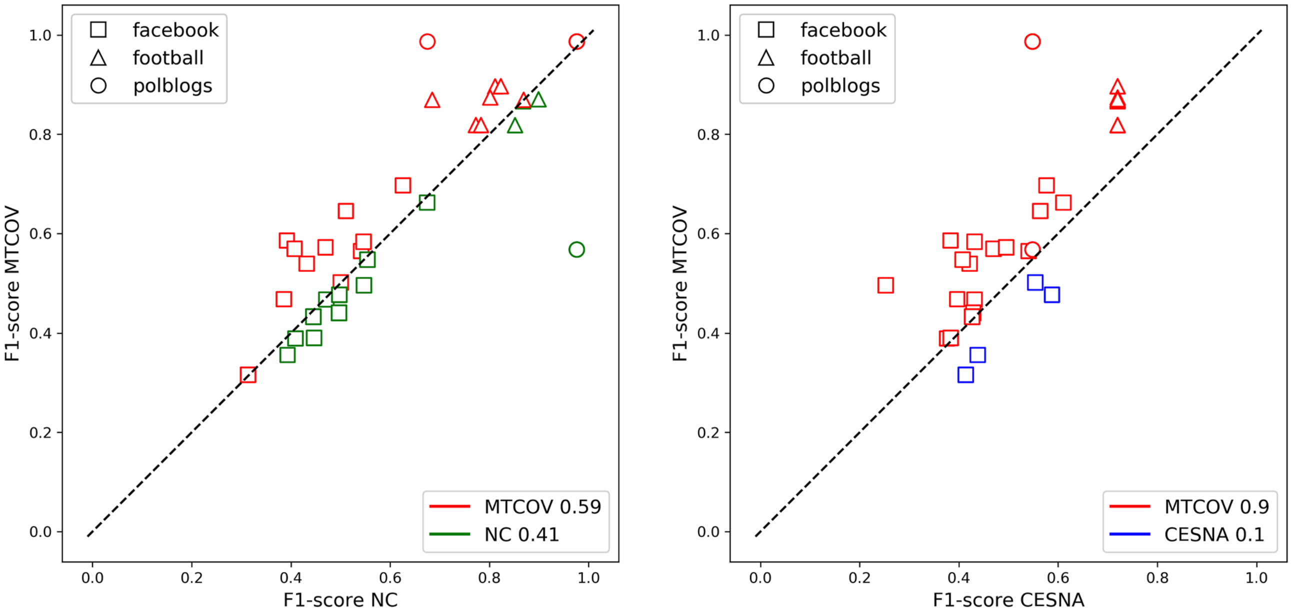

Figure 6. Trial-by-trial performance in F1-score. We compare MTCOV on the y-axis, with on the x-axis (left) NC and (right) CESNA. Markers denote the datasets: squares for *facebook*, triangles for *football* and circles for *polblogs*. Points above the diagonal, shown in red, are trials where MTCOV is more accurate than the other. The fractions for which each method is superior are shown in the plot legend.

Methods

We adapt recent ideas from the generative model behind MULTITENSOR[13], a multilayer mixed-membership model based on a Poisson tensor factorization[48], to incorporate node attributes in a principled manner. It can take in input directed and undirected networks, allowing different topological structures in each layer, including arbitrarily mixtures of assortative, disassortative and core-periphery structures. We move beyond MULTITENSOR by incorporating node covariates via introducing a proper likelihood term that accounts for this extra information. We use the formalism of maximum likelihood estimation: we combine the structural and the node information into a global likelihood function and provide a highly scalable Expectation-Maximization algorithm for the estimation of parameters.

Model description and notation. Consider a multilayer network of N nodes and L layers. This is a set of graphs $G = \{G^{(\alpha)}(\mathscr{V}, \mathscr{E}^{(\alpha)})\}_{1 \le \alpha \le L}$ defined on a set $\mathscr{V}$ of N vertices shared across $L \ge 1$ layers, and $\mathscr{E}^{(\alpha)}$ is the set of edges in the layer α. Each layer $\alpha \in \{1, \ldots, L\}$ is a graph $G^{(\alpha)}(\mathscr{V}, \mathscr{E}^{(\alpha)})$ with adjacency matrix $A^{(\alpha)} = [a_{ij}^{(\alpha)}] \in \mathbb{R}^{N \times N}$, where $a_{ij}^{(\alpha)}$ is the number of edges of type α from i to j; here we consider only positive discrete entries; for binary entries, $E = \sum_{i,j,\alpha} a_{ij}^{(\alpha)}$ is the total the number of edges. Alternatively, we can consider a 3-way tensor A with dimensions $N \times N \times L$. In addition, for each node $i \in \mathscr{V}$ consider the vector of covariates $X_i \in \mathbb{R}^{1 \times K}$ (alternatively called also attributes or metadata), where K is the total number of attributes. Here, for simplicity we focus on the case of $K = 1$ and categorical covariates with Z different categories. However, we can easily generalize to more than one covariate by encoding each possible combination of them as a different value of one single covariate. For example, for two covariates being gender and nationality, we can encode X_i being one covariate with possible values female/American, male/Spanish and so forth. One could also consider real-valued covariates by cutting them into bins. Nevertheless, a future expansion should include the possibility to work with any type of metadata.

A community is a subset of vertices that share some properties. Formally, each node belongs to a community to an extent measured by a C-dimensional vector denoted *membership*. Since we are interested in directed networks, for each node i we assign two such vectors, u_i and v_i (for undirected networks we set $u = v$); these determine how i forms outgoing and incoming links respectively. 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. Each community $k \in \{1, \ldots, C\}$ is linked to a category $z \in \{1, \ldots, Z\}$ by a parameter β_{kz}, that explains how much information of the z-th category is used to create the k-th community. To summarize, we consider two types of observed data: the adjacency tensor $A = \{A^{(\alpha)}\}_{1 \le \alpha \le L}$ and the design matrix $X = \{X_i\}_{i \in \{1, \ldots, N\}}$; the first contains information about the networks topology structure, the latter about the node covariates. In addition, we have the model parameters that we compactly denote as $\Theta = \{U, V, W, \beta\}$.

The goal is to find the latent parameters Θ using the data A and X. In other words, 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 $\forall i \in \{1, \ldots, N\}$; the affinity matrices $W^{(\alpha)}$, $\forall \alpha \in \{1, \ldots, L\}$, and the matrix $\beta = [\beta_{kz}] \in \mathbb{R}^{C \times Z}$, which captures correlations between communities and attributes. A visual overview of the proposed model is shown in Fig. 7. We consider a probabilistic generative model where MTCOV generates the network and the

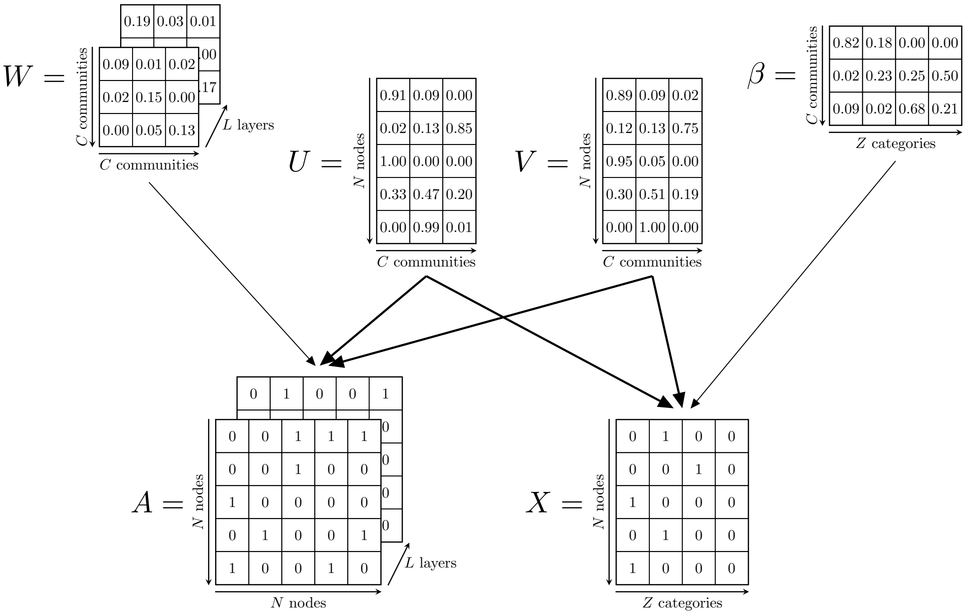

Figure 7. Graphical model representation of the algorithm MTCOV. A is the adjacency tensor, X is the design matrix and W, U, V, β are the latent parameters Θ. The membership matrices U and V couple the two datasets, and this is highlighted by the stronger thickness; whereas W and β are specific to the adjacency tensor and design matrix respectively. Here we present an example with binary adjacency matrix A, but the model is valid for more general weighted networks.

attributes probabilistically, assuming an underlying structure consisting of C overlapping communities. We adopt a maximum likelihood approach where, given the latent parameters Θ, we assume that the data A and X have independent likelihoods; in other words, we assume that A and X are *conditionally independent* given the latent parameters Θ. In addition, we assume that the memberships U and V couple the two datasets, as they are parameters shared between the two likelihoods; whereas the W and β are specific to the adjacency and design matrix respectively. We describe separately the procedures for modeling the topology of the network and the node attributes and then we show how to combine them in a unified log-likelihood framework.

Modeling the network topology.

In modeling the likelihood of the network topology, we adopt the ideas behind MULTITENSOR: we assume that the expected number of edges of type α from i to j is given by the parameter:

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

We then assume that each entry $a_{ij}^{(\alpha)}$ of the adjacency tensor is extracted from a Poisson distribution with parameter $M_{ij}^{(\alpha)}$. This is a common choice for network data[49–51] as it leads to tractable and efficient algorithmic implementations, compared for instance with other approaches that use Bernoulli random variables[9,27]; it also allows the flexibility of treating both binary and integer-weighted networks. We further assume that, given the memberships and affinity matrices, the edges are distributed independently; this is again a conditional independence assumption.

We can then write the likelihood of the network topology as:

$$P_G(A|U,V,W) = \prod_{i,j=1}^{N} \prod_{\alpha=1}^{L} \frac{e^{-M_{ij}^{(\alpha)}} \left(M_{ij}^{(\alpha)}\right)^{A_{ij}^{(\alpha)}}}{A_{ij}^{(\alpha)}!}, \tag{2}$$

which leads to the log-likelihood $\mathscr{L}_G(U,V,W)$ for the *structural dimension*:

$$\mathscr{L}_G(U,V,W) = \sum_{i,j,\alpha} \left[A_{ij}^{(\alpha)} \log \sum_{k,l} u_{ik} v_{jl} w_{kl}^{(\alpha)} - \sum_{k,l} u_{ik} v_{jl} w_{kl}^{(\alpha)} \right], \tag{3}$$

where we have neglected constants that do not depend on the parameters.

Modeling the node attributes.

In modeling the likelihood of the attributes, we assume that this extra information is generated from the membership vectors; this captures the intuition that knowing a node's community membership helps in predicting the value of the node's attribute. This assumption has also been made in other models for single-layer attributed networks[9] where one wants to enforce the tendency that nodes in the same community (for assortative structures) are likely to share common attributes. Different approaches[37,38] assume instead independence between attributes and membership, which follows a different idea of observing an interaction between individuals if either they belong to the same community (for assortative structures) or they share an attribute or both.

Then, we model the probability of observing the z-th category for the attribute covariate of node i as the parameter:

$$\pi_{iz} = \frac{1}{2} \sum_{k=1}^{C} \beta_{kz} (u_{ik} + v_{ik}), \tag{4}$$

where β_{kz} is the probability of observing a particular category z together with a community k; thus $\pi_i = (\pi_{i1} \ldots, \pi_{iZ})$ is a Z-dimensional vector such that $\pi_{iz} \in [0,1]$ and $\sum_{z=1}^{Z} \pi_{iz} = 1, \forall i$. For convenience, we consider one-hot encoding for $x_i = (x_{i1}, \ldots, x_{iZ})$, the realization of the random variable X_i: $x_{iz} = 1$ if node i has attribute corresponding to category z, 0 otherwise and $\sum_{z=1}^{Z} x_{iz} = 1$; the original design matrix $X_{N\times 1}$ is thus translated into a binary matrix $X_{N\times Z}$.

We then assume that each entry X_i of the design matrix is extracted from a multinomial distribution of parameter π_i, which yields the likelihood of the covariates:

$$P_X(X_i = x_i | U, V, \beta) = P_X(X_{i1} = x_{i1}, \ldots, X_{iZ} = x_{iZ} | U, V, \beta) = \pi_{i1}^{x_{i1}} \ldots \pi_{iZ}^{x_{iZ}} \ . \tag{5}$$

In order to satisfy the sum constraint $\sum_{z=1}^{Z} \pi_{iz} = 1$, we impose the normalizations $\sum_{z=1}^{Z} \beta_{kz} = 1$, valid $\forall k$ and $\sum_{k=1}^{C} u_{ik} = \sum_{k=1}^{C} v_{ik} = 1$, valid $\forall i$. Such constraints 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 and the memberships U and V. As done for the network's edges, we assume conditional independence for the attributes on the various nodes. This leads to the log-likelihood $\mathscr{L}_X(U, V, \beta)$ for the *attribute dimension*:

$$\mathscr{L}_X(U, V, \beta) = \sum_{i=1}^{N} \sum_{z=1}^{Z} x_{iz} \log(\pi_{iz}) = \sum_{i,z} x_{iz} \log\left(\frac{1}{2} \sum_{k} \beta_{kz} (u_{ik} + v_{ik})\right). \tag{6}$$

Note, we assume that the attributes have values that can be binned in a finite number Z of unordered categories and the attributes do not need to be one-dimensional. Indeed, we can encode each combination of more attributes as a different value of one-dimensional "super-attribute". The model will not be affected, but the computational complexity might increase.

Inference with the EM algorithm.

Having described how the model works and its main assumptions and intuitions, we now turn our attention to describe how to fit the parameters to the data, in other words, how to perform inference. We assume conditional independence between the network and attribute variables, thus we can decompose the total log-likelihood into a sum of two terms $\mathscr{L}(U, V, W, \beta) = \mathscr{L}_G(U, V, W) + \mathscr{L}_X(U, V, \beta)$. However, in practice, we can improve parameters' inference performance by better balancing the contributions of the two terms as their magnitude can be on different scales, thus the risk of biasing the total likelihood maximization towards one of the two terms. For this, we introduce a scaling parameter $\gamma \in [0,1]$ that explicitly controls the relative contribution of the two terms. The total log-likelihood is then:

$$\begin{aligned} \mathscr{L}(U, V, W, \beta) &= (1-\gamma)\mathscr{L}_G(U, V, W) + \gamma\, \mathscr{L}_X(U, V, \beta) \\ &= (1-\gamma) \sum_{i,j,\alpha} \left[A_{ij}^{(\alpha)} \log \sum_{k,l} u_{ik} v_{jl} w_{kl}^{(\alpha)} - \sum_{k,l} u_{ik} v_{jl} w_{kl}^{(\alpha)} \right] \\ &\quad + \gamma \sum_{i,z} x_{iz} \log\left(\frac{1}{2} \sum_{k} \beta_{kz} (u_{ik} + v_{ik})\right). \end{aligned} \tag{7}$$

Varying γ from 0 to 1 lets us interpolate between two extremes: analyzing the data purely in terms of the network topology or purely in terms of the attribute information. One can either fix this *a priori* based on the goal of the application, closer to 0 for link prediction or closer to 1 for attribute classification, or this can be treated as a hyperparameter that must be estimated, whose optimal value is obtained by fitting the data *via* tuning techniques (for instance cross-validation). This approach provides a natural quantitative measure for the dependence between the communities and the two sources of information. Notice that one can rescale *a priori* each likelihood term individually in order to control even more their magnitudes, and then add it to Eq. (7). This choice should be made based on the dataset at hand. Here we consider rescaling $\mathscr{L}_G$ and $\mathscr{L}_X$ only in studying the social support networks of Indian villages, as we have enough data for estimating the normalization coefficients; see Supplementary Section S3.1 for details.

We wish to find the $\Theta = (U, V, W, \beta)$ that maximizes Eq. (7). In general, this is computationally difficult, but we make it tractable by adopting a variational approach using an Expectation-Maximization (EM) algorithm[52], similar to what done by De Bacco et al.[13], but extended here to include attribute information. Namely, 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.

Using Jensen's inequality $\log \bar{x} \geq \overline{\log x}$ for each log-likelihood term gives:

$$\begin{aligned}\mathscr{L}_X(U,V,\beta) \geq \sum_{i,z} x_{iz} \sum_{k} h_{izk} \log \frac{\beta_{kz}(u_{ik}+v_{ik})}{2h_{izk}} &= \sum_{i,z,k} x_{iz}\left[h_{izk}\,\log \beta_{kz}(u_{ik}+v_{ik}) - h_{izk}\,\log 2h_{izk}\right] \\ &= \mathscr{L}_X(U,V,\beta,h)\end{aligned} \tag{8}$$

$$\mathscr{L}_G(U,V,W) \geq \sum_{i,j,k,l,\alpha} \left[A_{ij}^{(\alpha)}\left(\rho_{ijkl}^{(\alpha)}\,\log u_{ik}v_{jl}w_{kl}^{(\alpha)} - \rho_{ijkl}^{(\alpha)}\log \rho_{ijkl}^{(\alpha)}\right) - u_{ik}v_{jl}w_{kl}^{(\alpha)}\right] = \mathscr{L}_G(U,V,W,\rho). \tag{9}$$

These lower bounds hold with equality when

$$h_{izk} = \frac{\beta_{kz}(u_{ik}+v_{ik})}{\sum_{k'}\beta_{k'z}(u_{ik'}+v_{ik'})}, \quad \rho_{ijkl}^{(\alpha)} = \frac{u_{ik}v_{jl}w_{kl}^{(\alpha)}}{\sum_{k',l'} u_{ik'}v_{jl'}w_{k'l'}^{(\alpha)}}, \tag{10}$$

thus maximizing $\mathscr{L}_X(U,V,\beta)$ is equivalent to maximizing $\mathscr{L}_X(U,V,\beta,h)$; similarly for $\mathscr{L}_G(U,V,W)$ and $\mathscr{L}_G(U,V,W,\rho)$ (this was also the same result derived by De Bacco et al.[13]). Overall, we aim at maximizing $\mathscr{L}(U,V,W,\beta,h,\rho) = (1-\gamma)\mathscr{L}_G(U,V,W,\rho) + \gamma\,\mathscr{L}_X(U,V,\beta,h)$, in analogy with what was done before. These maximizations can be performed by alternatively updating a set of parameters while keeping the others fixed. The EM algorithm performs these steps by alternatively updating h, ρ (Expectation step) and Θ (Maximization step); this is done starting from a random configuration until $\mathscr{L}(\Theta,h,\rho)$ reaches a fixed point. Calculating Eq. (10) represents the E-step of the algorithm. The M-step is obtained by computing partial derivatives of $\mathscr{L}(\Theta,h,\rho)$ with respect to the various parameters in Θ and setting them equal to zero. We add Lagrange multipliers $\lambda = \left(\lambda^{(\beta)}, \lambda^{(u)}, \lambda^{(v)}\right)$ to enforce constraints:

$$\mathscr{L}'(\Theta,h,\rho,\lambda) = \mathscr{L}(\Theta,h,\rho) - \sum_k \lambda_k^{(\beta)}\left(\sum_{z=1}^{Z}\beta_{kz} - 1\right) - \sum_i \lambda_i^{(u)}\left(\sum_{k=1}^{C} u_{ik} - 1\right) - \sum_i \lambda_i^{(v)}\left(\sum_{k=1}^{C} v_{ik} - 1\right). \tag{11}$$

For instance, focusing on the update for β_{zk}, setting the derivative with respect to it in Eq. (11) to zero and enforcing the constraint $\sum_{z=1}^{Z}\beta_{kz} = 1$ gives $\lambda_k^{(\beta)} = \gamma\sum_{i,z} x_{iz}h_{izk}$; plugging this back finally gives:

$$\beta_{kz} = \frac{\sum_i x_{iz}\,h_{izk}}{\sum_{i,z} x_{iz}\,h_{izk}}, \tag{12}$$

which is valid for $\gamma \neq 0$. Doing the same for the other parameters yields (see Supplementary Section S1 for details):

$$u_{ik} = \frac{\gamma\,\sum_z x_{iz}h_{izk} + (1-\gamma)\,\sum_{j,l,\alpha} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}}{\gamma + (1-\gamma)\sum_{j,\alpha} A_{ij}^{(\alpha)}} \tag{13}$$

$$v_{ik} = \frac{\gamma\,\sum_z x_{iz}h_{izk} + (1-\gamma)\,\sum_{j,l,\alpha} A_{ji}^{(\alpha)}\rho_{jilk}^{(\alpha)}}{\gamma + (1-\gamma)\sum_{j,\alpha} A_{ji}^{(\alpha)}} \tag{14}$$

$$w_{kl}^{(\alpha)} = \frac{\sum_{i,j} A_{ij}^{(\alpha)}\rho_{ijkl}^{(\alpha)}}{\sum_i u_{ik}\sum_j v_{jl}} \quad , \tag{15}$$

where Eq. (15) is valid for $\gamma \neq 1$. The EM algorithm thus consists in randomly initializing the parameters Θ and then repeatedly alternating between updating h and ρ using Eq. (10) and updating Θ using Eqs. (12)–(15) until $\mathscr{L}(\Theta,h,\rho)$ reaches a fixed point. A pseudo-code is given in Algorithm 1. In general, the fixed point is a local maximum but we have no guarantees that it is also the global one. In practice, we run the algorithm several times, starting from different random initializations and taking the run with the largest final $\mathscr{L}(\Theta,h,\rho)$. The computational complexity per iteration scales as $O(M\,C^2 + NCZ)$, where M is the total number of edges summed across layers. In practice, C and Z have similar order of magnitude, usually much smaller than the system size M; for sparse networks, as is often the case for real datasets, $M \propto N$, thus the algorithm is highly scalable with a total running time linear in the system size. An experimental analysis of the computational time is provided in the Supplementary Section S2.

Notice that, although we started from a network log-likelihood $\mathscr{L}_G(U,V,W)$ similar to the one proposed in the MULTITENSOR model[13], the only update preserved from that is the one of w_{kl} in Eq. (15). The updates for u_{ik} and v_{ik} are instead quite different; the main reason is that here we incorporated the node attributes, which appear both explicitly and implicitly (through h) inside the updates. In addition, here we enforce normalizations like $\sum_k u_{ik} = 1$, not enforced in MULTITENSOR. This implies that our model restricted to $\gamma = 0$, i.e., no attribute information, does not correspond exactly to MULTITENSOR. This also implies that, upon convergence, we can directly interpret the memberships as *soft* community assignments (or overlapping) without the need of post-processing their values; in words, u_{ik} represent the probability of node i to belong to the *outgoing* community k, similarly for v_{ik} and an *incoming* membership. This distinction is necessary when considering directed networks. If one is interested in recovering *hard* memberships, where a node is assigned to only one community, then one can choose the community corresponding to the maximum entry of u or v.

```
**Algorithm 1** MTCOV- EM algorithm
**Input:** network $A = \{A_{ij}\}_{i,j=1}^{N}$, design matrix $X = \{x_{iz}\}_{i=1}^{N}$, number of communities $C$, hyperparameter $\gamma$
**Output:** membership vectors $U = [u_{ik}]$, $V = [v_{ik}]$; network-affinity matrix $W = [w_{kl}]$; attribute-affinity matrix $\beta = [\beta_{kz}]$.
Initialize $U, V, W, \beta$ at random.
Repeat until convergence:
  1. Calculate $h$ and $\rho$ (E-Step):
     $h_{izk} = \frac{\beta_{kz}(u_{ik}+v_{ik})}{\sum_{k'} \beta_{k'z}(u_{ik'}+v_{ik'})} \quad , \quad \rho_{ijkl}^{(\alpha)} = \frac{u_{ik} v_{jl} w_{kl}^{(\alpha)}}{\sum_{k',l'} u_{ik'} v_{jl'} w_{k'l'}^{(\alpha)}}$
  2. Update parameters $\Theta$ (M-Step):
     i) for each node $i$ and community $k$ update memberships:
        $u_{ik} = \frac{\gamma \sum_z x_{iz} h_{izk} + (1-\gamma) \sum_{j,l,\alpha} A_{ij}^{(\alpha)} \rho_{ijkl}^{(\alpha)}}{\gamma + (1-\gamma) \sum_{j,\alpha} A_{ij}^{(\alpha)}}$
        $v_{ik} = \frac{\gamma \sum_z x_{iz} h_{izk} + (1-\gamma) \sum_{j,l,\alpha} A_{ji}^{(\alpha)} \rho_{jilk}^{(\alpha)}}{\gamma + (1-\gamma) \sum_{j,\alpha} A_{ji}^{(\alpha)}}$
     ii) if $\gamma \neq 1$, for each pair of communities $(k,l)$ update network-affinity matrix:
        $w_{kl}^{(\alpha)} = \frac{\sum_{i,j} A_{ij}^{(\alpha)} \rho_{ijkl}^{(\alpha)}}{\sum_i u_{ik} \sum_j v_{jl}}$
     iii) if $\gamma \neq 0$, for each pair of community-attribute $(k,z)$ update attribute-affinity matrix:
        $\beta_{kz} = \frac{\sum_i x_{iz} h_{izk}}{\sum_{i,z} x_{iz} h_{izk}}$
```

Evaluation metrics. We adopt two different criteria for performance evaluation, based on having or not having access to ground-truth values for the community assignments. The first case applies to synthetic-generated data, the second to both synthetic and real-world data. We explain performance metrics in detail below.

Ground-truth available. In the presence of a known partition, we measure the agreement between the set of ground-truth communities $\mathscr{C}^*$ and the set of detected communities $\mathscr{C}$ using metrics for recovering both hard and soft assignments. For hard partitions, the idea is to match every detected community with its most similar ground-truth community and measure similarity $\delta(\mathscr{C}_i^*, \mathscr{C}_j)$ (and vice versa for every ground-truth community matched against a detected community) as done by Yang et al.[9]. The final performance is the average of these two comparisons:

$$\frac{1}{2|\mathscr{C}^*|} \sum_{\mathscr{C}_i^* \in \mathscr{C}^*} \max_{\mathscr{C}_j \in \mathscr{C}} \delta(\mathscr{C}_i^*, \mathscr{C}_j) + \frac{1}{2|\mathscr{C}|} \sum_{\mathscr{C}_j \in \mathscr{C}} \max_{\mathscr{C}_i^* \in \mathscr{C}^*} \delta(\mathscr{C}_i^*, \mathscr{C}_j), \tag{16}$$

where here we consider as similarity metric $\delta(\cdot)$ the F1-score and the Jaccard similarity.

In both cases, the final score is a value between 0 and 1, where 1 indicates the perfect matching between detected and ground-truth communities. For soft partitions, we consider two standard metrics for measuring distance between vectors as done by De Bacco et al.[13], such as *cosine similarity* (CS) and L_1 error, averaged over the nodes:

$$CS(U, U^0) = \frac{1}{N} \sum_{i=1}^{N} \frac{u_i \cdot u_i^0}{||u_i||_2 \, ||u_i^0||_2} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} \frac{u_{ik} \, u_{ik}^0}{||u_i||_2 \, ||u_i^0||_2} \tag{17}$$

$$L_1(U, U^0) = \frac{1}{2N} \sum_{i=1}^{N} ||u_i - u_i^0||_1 = \frac{1}{2N} \sum_{i=1}^{N} \sum_{k=1}^{C} |u_{ik} - u_{ik}^0|, \tag{18}$$

where u_i is the C-dimensional vector containing the i-th row of U, representing the detected membership and similarly for u_i^0 for the ground-truth U^0. The factor 1/2 ensures that the L_1 distance ranges from 0 for identical distributions to 1 for distributions with disjoint support. Similarly to the what done for hard partitions, we match the ground-truth and detected communities by choosing the permutation of C groups that gives the highest cosine similarity or smallest L_1 distance.

Ground-truth not available. In the absence of ground-truth, these metrics cannot be computed, and one must resort to other approaches for model evaluation. Here we consider performance in prediction tasks when hiding part of the input datasets while fitting the parameters, and in particular on the extent to which partial knowledge of network edges helps predict node attributes and vice versa. Thus we consider a measure for link-prediction and one for correct retrieval of the attributes. For link-prediction, we used the AUC statistic, equivalent to the area under the receiver-operating characteristic (ROC) curve[53]. It represents the probability that a randomly chosen missing connection (a true positive) is given a higher score 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 than chance. We measure the probability of observing an edge as the predicted expected Poisson parameters of Eq. (1). For the attribute, instead, we use the accuracy as a quality measure. For each node, we compute the predicted expected multinomial parameter π_i using Eq. (4). We then assign to each node the category with the highest probability, computing the accuracy as the ratio between the correctly classified examples over the total number of nodes. As baselines, we compare with the accuracy obtained with a random uniform probability and the highest relative frequency observed in the training set.

Cross-validation tests and hyperparameter settings. We perform prediction tasks using cross-validation with 80–20 splits: we use 80% of the data for training the parameters and then measure AUC and accuracy on the remaining 20% test set. Specifically, for the network topology, we hold out 20% of the triples (i, j, α); for the attributes, we hold out 20% of the entries of the categorical vector.

Our model has two hyperparameters, the scaling parameter γ and the number of communities C. We estimate them by using 5-fold cross-validation along with grid search to range across their possible values. We then select the combination $(\hat{C}, \hat{\gamma})$ that returns the best average performance over the cross-validation runs. Standard cross-validation considers performance in terms of a particular metric. However, here we have two possible ones which are qualitatively different, i.e., AUC and accuracy. Depending on the task at hand, one can define performance as a combination of the two, bearing in mind that the values of $(\hat{C}, \hat{\gamma})$ at the maximum of either of them might not coincide. Here we select $(\hat{C}, \hat{\gamma})$ as the values are jointly closer to both the maximum values. In the experiments where one of the two hyperparameters is fixed *a priori*, we run the same procedure but vary with grid search only the unknown hyperparameter.

Data availability

The code used for the analysis and to generate the synthetic data is publicly available and can be found at https://github.com/mcontisc/MTCOV.

Code availabilty

An open-source algorithmic implementation available at https://github.com/mcontisc/MTCOV.

Received: 28 April 2020; Accepted: 4 September 2020
Published online: 25 September 2020

References

1. Waskiewicz, T. Friend of a friend influence in terrorist social networks. In *Proceedings on the international conference on artificial intelligence (ICAI)*, 1 (The Steering Committee of The World Congress in Computer Science, Computer..., 2012).
2. Pinheiro, C. A. R. Community detection to identify fraud events in telecommunications networks. In *SAS SUGI proceedings: customer intelligence* (2012).
3. Pan, W.-F., Jiang, B. & Li, B. Refactoring software packages via community detection in complex software networks. *Int. J. Autom. Comput.* **10**, 157–166 (2013).
4. Bechtel, J. J. *et al.* Lung cancer detection in patients with airflow obstruction identified in a primary care outpatient practice. *Chest* **127**, 1140–1145 (2005).
5. Chen, J., Zhang, H., Guan, Z.-H. & Li, T. Epidemic spreading on networks with overlapping community structure. *Physica A Stat. Mech. Appl.* **391**, 1848–1854 (2012).
6. Traud, A. L., Kelsic, E. D., Mucha, P. J. & Porter, M. A. Comparing community structure to characteristics in online collegiate social networks. *SIAM Rev.* **53**, 526–543 (2011).
7. Newman, M. E. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**, 8577–8582 (2006).
8. Peel, L., Larremore, D. B. & Clauset, A. The ground truth about metadata and community detection in networks. *Sci. Adv.* **3**, e1602548 (2017).
9. Yang, J., McAuley, J. & Leskovec, J. Community detection in networks with node attributes. In *2013 IEEE 13th international conference on data mining*, 1151–1156 (IEEE, 2013).
10. Falih, I., Grozavu, N., Kanawati, R. & Bennani, Y. Community detection in attributed network. *Companion Proc. Web Conf.* **2018**, 1299–1306 (2018).

11. Fortunato, S. Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010).
12. De Domenico, M. *et al.* Mathematical formulation of multilayer networks. *Phys. Rev. X* **3**, 041022 (2013).
13. De Bacco, C., Power, E. A., Larremore, D. B. & Moore, C. Community detection, link prediction, and layer interdependence in multilayer networks. *Phys. Rev. E* **95**, 042317 (2017).
14. Schein, A., Paisley, J., Blei, D. M. & Wallach, H. Bayesian Poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining*, 1045–1054 (2015).
15. Schein, A., Zhou, M., Blei, D. M. & Wallach, H. Bayesian Poisson tucker decomposition for learning the structure of international relations. In *Proceedings of the 33rd international conference on machine learning*, vol. 48 (2016).
16. Valles-Catala, T., Massucci, F. A., Guimera, R. & Sales-Pardo, M. Multilayer stochastic block models reveal the multilayer structure of complex networks. *Phys. Rev. X* **6**, 011036 (2016).
17. Stanley, N., Shai, S., Taylor, D. & Mucha, P. Clustering network layers with the strata multilayer stochastic block model. *IEEE Trans. Netw. Sci. Eng.* **3**, 95–105 (2016).
18. Peixoto, T. P. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Phys. Rev. E* **92**, 042807 (2015).
19. Paul, S. *et al.* Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electron. J. Stat.* **10**, 3807–3870 (2016).
20. Gheche, M. E., Chierchia, G. & Frossard, P. Orthonet: multilayer network data clustering. *IEEE Trans. Signal Inf. Process. Netw.* **6**, 13–23 (2020).
21. Papadopoulos, A., Rafailidis, D., Pallis, G. & Dikaiakos, M. D. Clustering attributed multi-graphs with information ranking. In *Proceedings, Part I, of the 26th international conference on database and expert systems applications—volume 9261, DEXA 2015*, 432–446 (Springer, 2015).
22. Papadopoulos, A., Pallis, G. & Dikaiakos, M. D. Weighted clustering of attributed multi-graphs. *Computing* **99**, 813–840 (2017).
23. Chang, S. *et al.* Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '15*, 119–128 (2015).
24. Sachan, M., Contractor, D., Faruquie, T. A. & Subramaniam, L. V. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st international conference on world wide web, WWW '12*, 331–340 (2012).
25. Sweet, T. M. & Zheng, Q. Estimating the effects of network covariates on subgroup insularity with a hierarchical mixed membership stochastic blockmodel. *Soc. Netw.* **52**, 100–114 (2018).
26. Signorelli, M. & Wit, E. C. Model-based clustering for populations of networks. *Stat. Model.* **20**, 9–29 (2019).
27. Newman, M. E. & Clauset, A. Structure and inference in annotated networks. *Nat. Commun.* **7**, 11863 (2016).
28. Bothorel, C., Cruz, J. D., Magnani, M. & Micenkova, B. Clustering attributed graphs: models, measures and methods. *Netw. Sci.* **3**, 408–444 (2015).
29. Zhang, Y. *et al.* Community detection in networks with node features. *Electron. J. Stat.* **10**, 3153–3178 (2016).
30. Hric, D., Peixoto, T. P. & Fortunato, S. Network structure, metadata, and the prediction of missing nodes and annotations. *Phys. Rev. X* **6**, 031038 (2016).
31. Stanley, N., Bonacci, T., Kwitt, R., Niethammer, M. & Mucha, P. J. Stochastic block models with multiple continuous attributes. *Appl. Netw. Sci.* **4**, 1–22 (2019).
32. Emmons, S. & Mucha, P. J. Map equation with metadata: varying the role of attributes in community detection. *Phys. Rev. E* **100**, 022301 (2019).
33. Xu, Z., Ke, Y., Wang, Y., Cheng, H. & Cheng, J. A model-based approach to attributed graph clustering. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*, 505–516 (2012).
34. Bu, Z., Li, H.-J., Cao, J., Wang, Z. & Gao, G. Dynamic cluster formation game for attributed graph clustering. *IEEE Trans. Cybern.* **49**, 328–341 (2017).
35. Tallberg, C. A bayesian approach to modeling stochastic blockstructures with covariates. *J. Math. Sociol.* **29**, 1–23 (2004).
36. White, A. & Murphy, T. B. Mixed-membership of experts stochastic blockmodel. *Netw. Sci.* **4**, 48–80 (2016).
37. Airoldi, E. M., Choi, D. S. & Wolfe, P. J. Confidence sets for network structure. *Stat. Anal. Data Min. ASA Data Sci. J.* **4**, 461–469 (2011).
38. Sweet, T. M. Incorporating covariates into stochastic blockmodels. *J. Educ. Behav. Stat.* **40**, 635–664 (2015).
39. Taylor, D., Shai, S., Stanley, N. & Mucha, P. J. Enhanced detectability of community structure in multilayer networks through layer aggregation. *Phys. Rev. Lett.* **116**, 228301 (2016).
40. Taylor, D., Caceres, R. S. & Mucha, P. J. Super-resolution community detection for layer-aggregated multilayer networks. *Phys. Rev. X* **7**, 031056 (2017).
41. Holland, P. W., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: first steps. *Soc. Netw.* **5**, 109–137 (1983).
42. Power, E. A. *Building Bigness: Religious Practice and Social Support in Rural South India*. Doctoral Dissertation, Stanford University, Stanford, CA (2015).
43. Power, E. A. Social support networks and religiosity in rural South India. *Nat. Hum. Behav.* **1**, 0057 (2017).
44. Power, E. A. & Ready, E. Cooperation beyond consanguinity: post-marital residence, delineations of kin and social support among South Indian Tamils. *Philos. Trans. R. Soc. B Biol. Sci.* **374**, 20180070 (2019).
45. McAuley, J. & Leskovec, J. Learning to discover social circles in ego networks. In *Proceedings of the 25th international conference on neural information processing systems—volume 1, NIPS'12*, 539–547 (2012).
46. Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**, 7821–7826 (2002).
47. Adamic, L. A. & Glance, N. The political blogosphere and the 2004 U.S. election: divided they blog. In *Proceedings of the 3rd international workshop on link discovery, LinkKDD '05*, 36–43 (2005).
48. Kolda, T. G. & Bader, B. W. Tensor decompositions and applications. *SIAM Rev.* **51**, 455–500 (2009).
49. Ball, B., Karrer, B. & Newman, M. E. J. Efficient and principled method for detecting communities in networks. *Phys. Rev. E* **84**, 036103 (2011).
50. Gopalan, P. K. & Blei, D. M. Efficient discovery of overlapping communities in massive networks. *Proc. Natl. Acad. Sci. USA* **110**, 14534–14539 (2013).
51. Gopalan, P., Hofman, J. M. & Blei, D. M. Scalable recommendation with hierarchical poisson factorization. In *Proceedings of the 31-st conference on uncertainty in artificial intelligence*, 122–129 (2015).
52. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **39**, 1–22 (1977).
53. Hanley, J. A. & McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology* **143**, 29–36 (1982).

Acknowledgements

This work was partially supported by the Cyber Valley Research Fund. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Martina Contisciani. The authors are grateful for the goodwill of the residents of Teṉpaṭṭi and Aḻakāpuram, the support of faculty and students from the Folklore Department at Madurai Kamaraj University, and the assistance of the Chella Meenakshi Centre for Educational Research and Services. Funding for fieldwork was provided by the US National Science Foundation Doctoral Dissertation Improvement Grant (No. BCS-1121326), a Fulbright-Nehru Student Researcher Award, the Stanford Center for South Asia, and a National Science Foundation Interdisciplinary Behavioral & Social Science Research Grant (No. IBSS-1743019). We thank Cristopher Moore and Daniel Larremore for useful discussions and the Santa Fe Institute for providing the environment fostering these interactions.

Author contributions

M.C., E.P. and C.D.B. conceived the research and designed the analyses; M.C. conducted the experiment; all authors reviewed the manuscript.

Funding

Open Access funding provided by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41598-020-72626-y.

Correspondence and requests for materials should be addressed to C.D.B.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.