Community Detection in Large-Scale Networks: Literature Review

Aladin Djuhera

October 2020

This underlying document provides a summary of elementary theoretical principles in community detection. In particular, the general motivation, approach and further interesting deviations as well as extensions to the usual stochastic block model method are discussed.

Above principles are mostly derived from the following scientific publications:

- Network Analysis and Modeling (CSCI 5352), Prof. Aaron Clauset, University of Colorado Boulder
- Efficient and Principled Method for Detecting Communities in Networks
- Stochastic Blockmodels and Community Structure in Networks
- Structure and Inference in Annotated Networks

1 Network Analysis and Modeling: Large-Scale Stochastic Block Model Structures

1.1 Probabilistic General Models

Generative models are a more or less sophisticated random graph model approach. In general, they are defined by a parametric probability distribution $Pr(G|\theta)$ over all graphs, where G and θ denote the network instance and structural parameter set, respectively.

This probabilistic modeling approach allows for several advantageous features, such as:

- the definition of explicit parameters, which are interpretable w.r.t. the graph structure,
- a fair comparison of different approaches through likelihood scores,
- the assignment of probabilities to the observed structure, ultimately enabling the estimation of missing or future structures.

Furthermore, generation goes hand in hand with inference. Whereas the former one specifies a synthetic structure, inference allows for a statistical backpropagation to, for instance, such a generative model. In particular, inference in the sense of graph modeling assigns each pair of vertices i, j a probability value, which enters a pre-defined score function.

1.2 Stochastic Block Models

Stochastic Block Models (SBMs) represent the simplest form of generative models. They are conventionally defined for simple unweighted networks which exhibit distinctly non-overlapping communities within. However, they can be easily generalized to weighted and directed networks with so-called mixed memberships, that is overlapping community structures.

1.2.1 Model Definition

A SBM is defined by the tuple $\theta = (k, z, M)$:

- k: number of communities in the network. Note that for $k = 1$ the SBM becomes a simple Erdos-Renyi random graph model $G(n, p)$.
- z: $n \times 1$ vector, where $z_i \in 1, 2, ..., k$ denotes the group index of specified vertex i
- M: $k \times k$ stochastic block matrix, where M_{uv} denotes the probability that a vertex in group u is connected to a vertex in group v

Note that above parameter k needs to be specified *a priori*. In particular, we can regard k as the model order. As such, it becomes increasingly difficult to accurately estimate k a priori. Thus, most of these methods rely on a "hard code" definition of the number of communities. However, there also exist some very complicated and not yet fully universal approaches to estimate k beforehand.

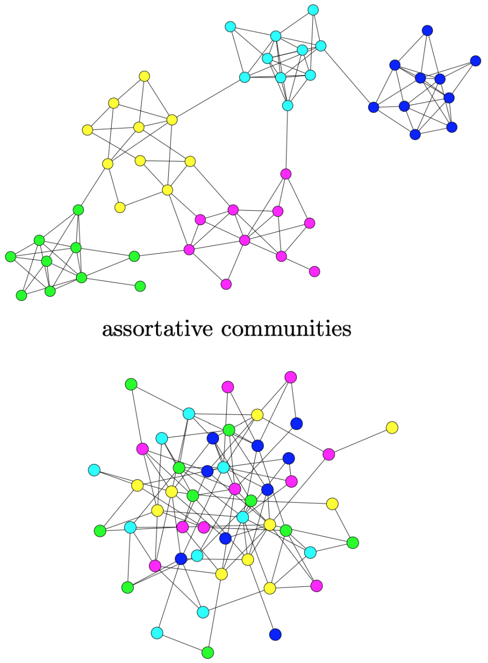
1.2.2 Network Generation

As soon as a parameter set θ is chosen, the network realization can be obtained from the SBM model by analyzing the stochastic block matrix M. For a distinct pair of vertices i, j, an edge between both exists with probability $M_{z_i z_j}$. As such, edges are independent but not identically distributed, ultimately being conditionally independent random variables.

1.2.3 Assortative and Disassortative Communities

In many publications, a clear distinction between assortative and disassortative relationships is made.

Assortative communities are ones where vertices tend to connect to alike vertices, that is vertices that oftentimes lie in the same community as they exhibit similar properties. In such a case, M has a distinct pattern for which the diagonal elements are strictly greater than the values off the diagonal. In particular, it holds $M_{uu} > M_{uv}$ for $u \neq v$. Disassortative communities, on the other hand, are the exact opposite, where connections between "unsimilar" vertices are more likely and for which it holds $M_{uu} < M_{uv}$ for $u \neq v$. A possible realization of assortative and disassortative communities is given in below figure.



assortative communities

disassortative communities

Figure 1: Assortative and Disassortative Community Realizations

1.2.4 Degree Heterogeneity

In general, SBMs generate networks which are of random graph model character. As such, the degree distribution of the respective vertices is always Poisson distributed and thus more or less homogenous. However, recent studies (especially by Karrer and Newman) have found that in real-world networks, vertices only rarely underly such a distribution. In fact, a significantly higher performance, for both real-world and synthetic network structures, can be achieved when incorporating varying vertex degrees, that is a heterogenous degree distribution.

1.2.5 SBM Fit to Data

In order to fit the observed data from the underlying graph G to the SBM model, a maximum likelihood approach is pursued. In particular, the parameters z and M are chosen to maximize the likelihood of generating G via the SBM structure. The respective likelihood can be denoted as

$$\begin{aligned}\mathcal{L}(G|M,z) &= \prod_{i,j} Pr(i \to j|M,z) \\ &= \prod_{(i,j)\in E} Pr(i \to j|M,z) \quad \prod_{(i,j)\notin E} 1 - Pr(i \to j|M,z) \\ &= \prod_{i,j\in E} M_{z_i,z_j} \quad \prod_{i,j\notin E} (1 - M_{z_i,z_j}) \qquad (1)\end{aligned}$$

where edges we observe are denoted by $(i,j) \in E$ and those we do not by $(i,j) \notin E$. Furthermore, let N_u count the number of vertices with label u and let $N_{uv} = N_u N_v$ be the number of possible edges for $u \neq v$.

Also, assume that for a particular vertex labeling z and a corresponding choice of groups u and v, we observe E_{uv} edges between u and v. Since E_{uv} is binomially distributed, the corresponding maximum likelihood choice for the probability M_{uv} is simply the maximum likelihood estimate for a binomial with expected value E_{uv}. Thus, we obtain the respective estimate

$$\hat{M}_{uv} = \frac{E_{uv}}{N_{uv}} \qquad (2)$$

Above expression can now be simplified considerably into the following relation:

$$\begin{aligned}\mathcal{L}(G|M,z) &= \prod_{u,v} M_{uv}^{E_{uv}} (1 - M_{uv})^{N_{uv}-E_{uv}} \qquad (3) \\ &= \prod_{u,v} (\frac{E_{uv}}{N_{uv}})^{E_{uv}} (1 - \frac{E_{uv}}{N_{uv}})^{N_{uv}-E_{uv}} \qquad (4)\end{aligned}$$

Furthermore, by taking the logarithm of above expression, we do not change the statistical properties. As such, we can apply **Expectation Maximization (EM)** techniques to find the optimal parameter set θ.

2 Efficient and Principled Method for Detecting Communities in Networks

2.1 What is Community Detection?

- Detection of groups of densely interconnected nodes
- Nodes may be overlapping or disjoint

2.2 What is the Aim of the Paper?

- Communities are being detected based on a principled statistical approach using generative network models
- Method is implemented using a fast, closed-form expectation-maximization algorithm
- Network analysis of millions of nodes is possible with reasonable runtimes
- Works with overlapping and non-overlapping communities
- Shown for undirected, unweighted networks

2.3 Introduction

- Requirements for good detection: Effictive (recognize community structures when they are present), Fundamental (based on a mathematically or physically sound principle which renders it more trustworthy), Fast and Scalable w.r.t. computational efforts
- Demonstrated algorithm is based on ML and EM algorithms
- Developed for overlapping communities which can be easily expanded to non-overlapping ones

In general, the algorithm is a global statistical method for detecting overlapping communities based on the idea of link communities.

- Communities arise when there are links (different types of edges) in a network
- If we can identify the links and interconnections, we can deduce them to the respective vertices
- Overlapping communities can then be identified when vertices have more than one type of link

The paper demonstrates community detection by first defining the community model and fitting the best values of its parameters using an ML algorithm which effectively identifies overlapping community structures. By assigning each vertex solely to the community mto which it most strongly belongs to, the link community model can be seen as a relaxation of a stochastic block model which eventually is able to detect non-overlapping communities.

2.4 Generative Model for Link Communities

- First define the generative network model with n vertices and undirected edges divided by a number K of communities
- The model is defined by its paramters θ_{iz} (propensity of vertex i to have edges of color z) and the product $\theta_{iz}\dot{\theta}_{jz}$ which is the expected (poisson distribution assumed) number of edges of color z between vertices i and j
- Model is a multigraph and allows self-edges

2.5 Detecting Overlapping Communities

- The probability for the generation of a graph G with adjacency matrix elements A_{ij} is given by the following Poisson distribution

$$P(G|\theta) = \prod_{i<j} \frac{(\sum z\theta_{iz}\theta_{jz})^{A_{ij}}}{A_{ij}!} \exp - \sum_{z} \theta_{iz}\theta_{jz} \times \prod_{i} \frac{(\frac{1}{2}\sum_z \theta_{iz}\theta_{iz})^{\frac{A_{ii}}{2}}}{\frac{A_{ii}}{2}!} \exp - \frac{1}{2}\sum_{z} \theta_{iz}\theta_{iz} \tag{5}$$

- The model is eventually fit to an observed network by maximizing above logarithmic probability w.r.t. the parameters θ_{iz}

$$\log P(G|\theta) = \sum_{ij} A_{ij} \log(\sum_{z} \theta_{iz}\theta_{jz}) - \sum_{ijz} \theta_{iz}\theta_{jz} \tag{6}$$

- Direct maximization is hard due to nonlinear implicit equations, therefore an expectation-maximization approach is pursued where the below two equations are simultaneously solved. In order to prevent a local maximum optimization, the process is repeated several times using different random initializations and the maximum final log likelihood set is eventually chosen.

$$q_{ij}(z) = \frac{\theta_{iz}\theta_{jz}}{\sum_z \theta_{iz}\theta_{jz}} \tag{7}$$

$$\theta_{iz} = \frac{\sum_j A_{ij}q_{ij}(z)}{\sqrt{\sum_{ij} A_{ij}q_{ij}(z)}} \tag{8}$$

Here, q_{ij} simply denotes the probability that an edge between i and j has the color z which is exactly the quantity we need in order to infer link communities in the respective network. Furthermore, it is clear that q_{ij} is symmetric for undirected networks.

2.6 Implementation

- Rather than focusing on the actual parameter set θ_{iz}, it is worked with the average number k_{iz} of ends of edges of color z connected to the vertex i and the average number κ_z of edges of color z summed over all vertices which eventually leads to θ_{iz} in below relation.

$$k_{iz} = \sum_{j} A_{ij}q_{ij}(z) \tag{9}$$

$$\kappa_z = \sum_{i} k_{iz} \tag{10}$$

$$\theta_{iz} = \frac{k_{iz}}{\sqrt{\kappa_z}} \tag{11}$$

- The running time now efficiently reduces to $O(mK)$ where m is the number of edges in the network and K represents the number of possible communities. The process speed can be further improved by applying pruning techniques to the respective k_{iz} whose computation can be discarded once the value drops below a certain bound delta.

2.7 Conclusion

- Successful application to large-scale overlapping and non-overlapping communities
- Fast implementation
- Drawbacks: No criterion for determining the value of K (number of communities). The authors believe that statistical model selection methods applied to generative models should in principle lead to good results, though this approach is computationally very demanding.

3 Stochastic Blockmodels and Community Structure in Networks

3.1 What is the Purpose of the Paper?

- Stochastic blockmodels are not only qualified for detecting community structures in networks but also suffice to generate synthetic networks
- Synthetic networks can be used for several benchmarks
- The paper adresses the real-world problem of varying vertex degrees which are usually not considered by traditional community detectors
- It shows that a significant increase in performance can be achieved when incorporating varying vertex degrees for both real-world and synthetic network structures

3.2 Introduction

- Stochastic blockmodels belong to the class of random graph models. They represent a generative model approach for blocks, groups or communities and networks. In its simplest form, they assign vertices n to K blocks/groups/communities. Their respective relations can be represented as undirected edges which are a function of the vertex group memberships.
- The most important application of SBMs is the fitting to empirical network data in order to discover/identify community structures. This process is also denoted as a posteriori blockmodeling.
- In general, such an a posteriori fitting gives bad results as the simple SBM is usually not flexible enough to cover real-world scenarios.
- More complex modeling approaches, which aim to tackle above disadvantages of the simple SBM, usually result in heavy modifications, generally diminishing analytic interpretability. The paper thus suggests incorporating a heterogeneity in vertex degrees which in fact leads to substantial advantages.
- In general, the paper proposes a degree-corrected model with a closed-form parameter solution which is only slightly more complex than a simple general SBM. Its idea can be easily applied to other blockmodel approaches such as the overlapping of mixed-membership models.

3.3 Degree-Corrected Stochastic Blockmodel

- In the corrected version, the probability distribution does not only depend on the general parameters but also on a new parameter set θ_i which controls the expected degrees of vertices i.
- Incorporating such a structure into the general (simple) SBM and letting the expected value of the adjacency matrix element A_{ij} be $\theta_i\theta_j w_{g_i g_j}$ leads to the probability of graph G

$$P(G|\theta, w, g) = \prod_{i<j} \frac{(\theta_i\theta_j w_{g_i g_j})^{A_{ij}}}{A_{ij}!} \exp(-\theta_i\theta_j w_{g_i g_j}) \times \prod_i \frac{(\frac{1}{2}\theta_i^2 w_{g_i g_i})^{\frac{A_{ii}}{2}}}{(\frac{A_{ii}}{2})!} \exp(-\frac{1}{2}\theta_i^2 w_{g_i g_i}) \tag{12}$$

- Using this relation alongside other substitutions eventually leads to a more compact unnormalized log likelihood function which only differs from the general uncorrected SBM in the replacement of the number n_r of vertices in each group by the number κ_r of stubs (total number of ends of edges). This very replacement indeed leads to a huge difference in outcome.

$$\mathcal{L}(G|g) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{\kappa_r \kappa_s} \tag{13}$$

3.4 Conclusion

- The degree-corrected SBM versions can be used to generate synthetic network structured which retain the generality and tractability of other blockmodels while producting degree sequences closer to those of real-worls networks.
- The degree-corrected SBMs perform significantly better to both synthetic and real-world networks (i.e. fitting). Uncorrected versions tend to split the network into groups of high and low degree, eventually preventing it from finding the true group memberships. The degree-corrected model correctly ignores such divisions based solely on degree and is thus more sensitive to the underlying structure itself.
- More sophisticated blockmodel structures such as overlapping or mixed membership models would certainly benefit from such a degree heterogeneity incorporation.
- Degree-corrected models suffer from some drawbacks:
    - Can produce an unrealistic number of zero-degree vertices
    - Unable to model some degree sequences such as thos in which certain values of the degree are entirely forbidden
    - Might fail to accurately represent higher-order network structures such as overrepresented network motifs or degree correlations
    - Number of model parameters scales with the size of the network which might prevent fits to a network of one size being used to generate synthetic networks of another size
    - As always, the number K of blocks/groups/communities is hard to estimate and oftentimes we rely on a given value

4 Structure and Inference in Annotated Networks

4.1 What Distinguishes this Paper's Approach from other Detection Methods?

- In general, community detection is pursued in a two-fold manner here: Incorporating network and metadata characteristics.
- Whilst the previous papers focused solely on the network parameters, incorporating individual metadata such as gender, age, etc. in a social network realization for example, allows for a more accurate detection.
- The suggested method does not assume a correlation between network parameters and metadata in the first place, instead, it learns whether a correlation exists and if so correctly uses or ignores the metadata depending on whether they contain any useful information.
- Metadata can be represented on the nodes or edges either way. Furthermore, it can be of categorical or real-valued character.
- Community detection is also referred to as node clustering or node classification.

4.2 Introduction

- Usually, networks represent an assortative structure for which denser connections lie within the community/group itself and more sparse ones connect the individual groups to each other. However, disassortative structures can also occur.
- In some cases, the respective groups correlate meaningfully with certain metadata properties. However, this is rather the exception than the rule. In particular, not all metadata properties might be relevant.
- The resulting adapted methods have several attractive features: First, metadata is used in arbitrary format to improve the accuracy of community detection. Second, they do not assume a priori that the metadata correlated with the communities. Instead they detect and quantify the relationship between metadata and community if one exists and then exploit the relationship to improve results. Even with noisy relations, some information can be retrieved for better predictions. For no correlation at all, the method simply ignores the metadata contribution. Third, the methods allow for a more thoughtful selection between competing network divisions. This means that if we know that the community can be somewhat distinguished by member age and if we have metadata information of some of the nodes, we can use this information to steer the algorithm towards an age-correlated division technique.
- In particular, once a correlation between the metadata and the network structure is found, this ultimately yields a very interpretable relation which allows to predict memberships for nodes of which we do not have enough network data but only metadata, for example.
- The discussed approach takes as input a network accompanied by a set of node metadata. The respective output is a division of the nodes of the network into a specified number k of groups or communities.
- The method does not assume a variation in density (more dense within, less dense between the groups), is numerically efficient and makes use of a so-called belief propagation scheme for rapid inference of optimal group assignments. The latter one ultimately leads to an application to very large networks $\sim$ 1.4 million nodes.

4.3 Further Extensions to the Method

- Extension of the metadata: Include more complex metadata formats such as combinations of discrete and coninuous variables, multi-dimensional vectors, etc.
- Use metadata to identify network structures such as hierachies, motifs, rankings, latent-space structures, etc.

4.4 Methodology

- As usual, Bayesion statistical inference methods are applied for which a generative network model is fitted to an observed network with accompanying metadata. The resulting parameters of the fit eventually yield information about the network structure.
- The paper suggests a modified SBM with the following modifications: First, a modification is applied for which a more heterogenous distribution of the node degrees is respected (as discussed in above paper). This is done by including a degree-correction term which matches the modelled node degrees to the observed data. Second, a dependence on node metadata via a set of prior probabilities is introduced. This prior probability of a node belonging to a particular community then becomes a function of the metadata and this exact function is learned by the algorithm and ultimately incorporates the metadata into the calculation.
- A distinction is made between ordered and unordered metadata.

4.4.1 Unordered Data Approach

- Consider an undirected network with n nodes labeled by integers $u = 1, ..., n$ divided among k communities.
- The community to which node u belongs to is denoted by $s_u \in 1, ..., k$.
- The metadata is considered to be a finite set K of discrete, unordered values x_u for the respective node $u \in 1, ..., K$.
- The respective network can then be generated using the metadata $x = x_u$ and degree $d = d_u$ for all nodes.
- First, each node is assigned to a a community s with a probability depending on the metadata.
- The probability of assignment is denoted as γ_{sx} for each combination s, x of community and metadata.
- The prior probability on community assignments is thus given by below expression, where Γ is a $k \times K$ matrix of parameters y_{sx}.

$$P(s|\Gamma, x) = \prod_i \gamma_{s_i, x_i} \tag{14}$$

- Once every node has been assigned, the respective edges are placed with a probability of an edge between nodes u and v p_{uv}. Note that in below equation θ is an index-symmetric parameter set and $d_u d_v$ allows the model to fit arbitrary degree sequences.

$$p_{uv} = d_u d_v \theta_{s_u, s_v} \tag{15}$$

- Community detection then proceeds as usual w.r.t. an $n \times n$ adjacency matrix A. The probability of fit is given by $P(A|\Theta, \Gamma, x)$ with Θ being a $k \times k$ matrix with elements θ_{st}. The sum eventually represents all possible community assignments.

$$\begin{aligned} P(A|\Theta, \Gamma, x) &= \sum_s P(A|\Theta, s) P(s|\Gamma, x) \\ &= \sum_s \prod_{u<v} p_{uv}^{a_{uv}} (1 - p_{uv})^{1 - a_{uv}} \prod_u \gamma_{s_u, x_u} \end{aligned} \tag{16}$$

- Proceeding with the EM strategy, we maximize the logarithmic expression of above relation:

$$\log P(A|\Theta, \Gamma, x) = \log \sum_s P(A|\Theta, s) P(s|\Gamma, x) \tag{17}$$

- Furthermore, making use of Jensen's inequality, for which any set of positive quantities x_i summed in the logarithm obey below relation where q_i is any correctly normalized probability distribution such that $\sum_i q_i = 1$.

$$\log \sum_i x_i \geq \sum_i q_i \log \frac{x_i}{q_i} \tag{18}$$

- Once again, exact equality is achieved by the particular choice

$$q_i = \frac{x_i}{\sum_i x_i} \tag{19}$$

- Applying Jensen's equality to above logarithmic expression in (17) yields

$$\begin{aligned} \log P(A|\Theta, \Gamma, x) &\geq \sum_s q(s) \log \frac{P(A|\Theta, s)P(s|\Gamma, x)}{q(s)} \\ &= \sum_s q(s) \log P(A|\Theta, s) + \sum_s q(s) \log P(s|\Gamma, x) - \sum_s q(s) \log q(s) \end{aligned} \tag{20}$$

- Note that $q(s)$ is any valid distribution over community assignments s such that the sum equals 1, i.e. $\sum_s q(s) = 1$. The respective maximum of the above right-hand side coincides with the exact equality when it holds

$$q(s) = \frac{P(A|\Theta, s)P(s|\Gamma, x)}{\sum_s P(A|\Theta, s)P(s|\Gamma, x)} \tag{21}$$

- Thus, a double maximization is needed and repeatedly applied in order to eliminate local minima. The algorithm is as follows.
  - Make an initial guess about the parameter values and use them to calculate the optimal $q(s)$ from (21).
  - Using that value, maximize the right-hand side of (20) with respect to the parameters, while holding $q(s)$ constant.
  - Repeat from step 1 until convergence is achieved

- The exact calculation is given in the paper. However, maximization tells us that once the iteration converges, we obtain

$$q(s) = \frac{P(A|\Theta, s)P(s|\Gamma, x)}{\sum_s P(A|\Theta, s)P(s|\Gamma, x)} = \frac{P(A, s|\Theta, \Gamma, x)}{P(A|\Theta, \Gamma, x)} = P(s|A, \Theta, \Gamma, x) \tag{22}$$

- In particular, $q(s)$ is the so-called posterior distribution over community assignments.

- Eventually, q_s^u tells us the division, that is to which group the node belongs. The prior probability y_{sx} tells us to how and to what extent the metadata are correlated with the communities. For no correlation, the prior probabilities become constant and have no influence on the posterior probabilities.

- Since the EM maximization of above $q(s)$ denominator expression is intractable for large networks due to exponentially growing parameters, Monte Carlo importance sampling can be performed to approximate the distribution $q(s)$. Alternatively, the method hier proposes a belief propagation which is significantly faster and fast enough for practical applications.

4.4.2 Ordered Data Approach

- For ordered metadata and potentially continuous variables such as age or income in a social network, a different algorithm is required. The prior probabilitie $P(s|x)$ of belonging to community s given metadata value x now becomes a continuous function of x.
- In order to remain interpretability, arbitrary changes of values of $P(s|x)$ cannot be allowed. Thus, the expression should be a smooth functional.
- Using Bernstein polynomials, the degree of the functional can be reasonably set. The prior probabilities of the EM algorithm can then be expressed as

$$P(s|\Gamma, x) = \prod_u P(s_u|x_u) \tag{23}$$

- The optimal coefficients y_{sj} of the optimal degree-N polynomial prior can then be obtained by iterating below two expressions until convergence

$$Q_j^{su} = \frac{\gamma_{sj} B_j(x_u)}{\sum_k \gamma_{sk} B_k(x_u)} \tag{24}$$

$$\tag{25}$$

$$\gamma_{sj} = \frac{\sum_s q_s^u Q_j^{su}}{\sum_{tu} q_t^u Q_j^{tu}} \tag{26}$$

4.5 Normalized Mutual Information (NMI)

- A very important measure of quality is the so-called normalized mutual information.
- It measures to level of agreement between community divisions and 'ground truth' variables. Given a community division represented by an n-element vector s of group labels and discrete metadata represented by x, the conditional entropy of the community division is

$$H(s|x) = -\sum_x P(x) \sum_s P(s|x) \log P(s|x) \tag{27}$$

- $P(x)$ is the fraction of nodes with metadata x and $P(s|x)$ is the probability that a node belongs to a community s if it has metadata x.
- Traditionally the logarithm is taken in base 2, in which case the units of conditional entropy are bits. The conditional entropy is equal to the amount (in bits) of additional information one would need, on top of the metadata themselves, to specify the community membership of every node in the network. If the metadata are perfectly correlated with the communities, so that knowing the metadata tells us the community of every node, then the conditional entropy is zero. Conversely, if the metadata are worthless, telling us nothing at all about community membership, then the conditional entropy takes its maximum value, equal to the total entropy of the community assignment $H(s) = -\sum_s P(s) \log P(s)$. In our case we already know the value of $P(s|x)$: it is equal to the prior probability γ_{sx} of belonging to community s, one of the outputs of the algorithm. Hence

$$\begin{aligned} H(s|x) &= -\sum_x P(x) \sum_s \gamma_{sx} \log \gamma_{sx} = -\sum_x \frac{n(x)}{n} \sum_s \gamma_{sx} \log \gamma_{sx} \\ &= -\frac{1}{n} \sum_{su} \gamma_{s,x_u} \log \gamma_{s,x_u} \end{aligned} \tag{28}$$

- Note that $n(x) = nP(x)$ is the number of nodes with metadata x and n is the total number of nodes in the network, as previously.
- Alternatively, if we want a measure that increases (rather than decreases) with the amount of information the metadata give us, we can subtract $H(s|x)$ from $H(s)$, which gives the (unnormalized) mutual information

$$I(s, x) = H(s) - H(s|x) \tag{29}$$

- This quantity has a range from zero to $H(s)$, making it potentially hard to interpret, so commonly one normalizes it, creating the normalized mutual information. There are several different normalizations in use. As discussed by McDaid et al., it is mathematically reasonable to normalize by the larger, the smaller or the mean of the entropies $H(s)$ and $H(x)$ of the communities and metadata. Danon et al. originally used the mean, while Hric et al. in their work on lack of correlation between communities and metadata used the maximum. In the present case, however, we contend that the best choice is the minimum.

- The largest possible value of the mutual information is $H(s)$, which sets the scale on which the mutual information should be considered large or small. Thus, one might imagine the correct normalization would be achieved by simply dividing $I(s, x)$ by $H(s)$, yielding a value that runs from zero to one. This, however, would give a quantity that was asymmetric with respect to s and x —if the values of the two vectors were reversed the value of the mutual information would change.

- Mutual information, by convention, is symmetric and we would prefer a symmetric definition. Dividing by $min[H(s), H(x)]$ achieves this. In all the examples we consider, the number of communities is less than the number of metadata values, in some cases by a wide margin. Assuming the values of both to be reasonably broadly distributed, this implies that the entropy $H(s)$ of the communities will be smaller than that of the metadata $H(x)$ and hence $min[H(s), H(x)] = H(s)$. Thus if we define the NMI as follows, we ensure that the normalized mutual information lies between 0 and 1, that it has a symmetric definition with respect to s and x, and that it will achieve its maximum value of one when the metadata perfectly predict the community membership. Other definitions, normalized using the mean or maximum of the two entropies, satisfy the first two of these three conditions but not the third, giving values smaller than one by an unpredictable margin even when the metadata perfectly predict the communities.

$$NMI = \frac{I(s,x)}{min[H(s), H(x)]} \tag{30}$$