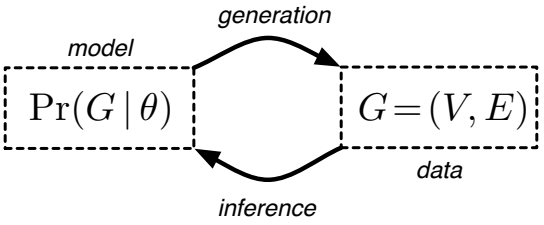
1 Inferring large-scale structural patterns

A powerful alternative to analyzing and modeling large-scale patterns in networks is probabilistic *generative models*, which are a sophisticated form of random graph model. Under this approach, we define a parametric probability distribution over all graphs $\Pr(G \mid \theta)$ in which θ encodes certain structural patterns within an ensemble of networks and $\Pr(G \mid \theta)$ specifies how often we expect to see these patterns.

Given a particular choice of θ (or a means for generating such a choice), we can generate a network instance G from the distribution by flipping a set of coins whose biases are controlled by θ. Inference is the reverse of this process: we are given some network G, whether generated synthetically from the model or obtained empirically, and we aim to identify the best choice of θ for generating it.



Like other approaches to analyzing the large-scale structure of observed networks, i.e., to extracting a good coarse-graining of the network, inference with generative models relies on a score function to decide which parameterizations are good choices. Unlike other approaches, including modularity Q, generative models explicitly assign a probability value to each pair i, j in the network.[1] As a result, generative models are a powerful method for encoding specific assumptions about the way "latent" or unknown parameters interact to create edges, and offer many advantageous features. For example,

- they make our assumptions about network structure explicit (rather than encoding them within a procedure or algorithm),
- their parameters can often be interpreted relative to specific structural hypotheses,
- they facilitate the fair comparison of alternative models via their likelihood scores,
- they enable the generative of synthetic data with specified structural patterns, which can be used to evaluate the model's goodness of fit, as a substrate for network simulations, and more,
- they assign probabilities to the observation or not of specific network features, and

[1] Most generative models for networks are what you might call "edge generative," meaning that they do not consider networks with different numbers of vertices, only networks of fixed size with different patterns of edges.

- they enable the estimation of *missing* or *future* structures, based on a partial or past observations of network structure.

These benefits do come at some cost. For instance, both specifying and fitting a particular model to the data can seem more complicated than with simple heuristic approaches or vertex-/network-level measures. In practice, however, generative models are often worth the additional effort as a result of the statistical rigor that they bring to the general task of network analysis and modeling.

2 The stochastic block model

The simplest generative model for networks is the *stochastic block model* (or sometimes "stochastic blockmodel," both abbreviated SBM). This model was first studied in mathematical sociology[2] in the mid-1980s and is now commonly used in network analysis in machine learning, complex systems, and statistical physics.

Conventionally, the SBM is defined for simple unweighted networks and for non-overlapping community assignments. It is easily generalized to directed networks and those with self-loops. Versions with weights, overlapping ("mixed") memberships, arbitrary degree distributions, and other features have also been introduced.

2.1 Model definition

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

- k : a scalar value denoting the number of groups (or modules or communities) in the network,
- z : a $n \times 1$ vector where $z_i \in \{1, 2, \ldots, k\}$ [or $z(i)$] gives the group index of vertex i,
- M : a $k \times k$ stochastic block matrix, where M_{uv} gives the probability that a vertex in group u is connected to a vertex in group v.

Of these parameters, k is special because it must be chosen before either z or M can be written down. Given these choices, every pair i, j is assigned some probability of being connected. Every vertex has a group or "type" assignment given by z, and knowing the assignments z_i and z_j allows us to index into the matrix M to find the probability that such an edge exists.

[2]First introduced in Holland, Laskey, and Leinhardt, "Stochastic blockmodels: First steps." *Social Networks* **5**(2), 109–137 (1983), and further developed in Wang and Wong, "Stochastic Blockmodels for Directed Graphs." *J. American Statistical Association* **82**(397), 8–19 1987.

The central assumption of the SBM is thus:

vertices within a group connect to other groups according to their group membership alone.

In this way, vertices in the same group are *stochastically equivalent* because they have equivalent connectivity patterns to other vertices. That is, every vertex in group u has the same set of probability values that govern the connections to other vertices, and this set is given by the uth row (or column) of the matrix M.

2.2 Generating networks with the SBM

Given a choice of the tuple (k, z, M), we can draw a network instance from the SBM model by flipping a coin for each pair of vertices i, j where that edge exists with probability $M_{z(i),z(j)}$. In this way, edges are independent but not identically distributed. Thus, edges are conditionally independent random variables, i.e., conditioned on their types, all edges independent, and for a given pair of types u, v, edges are iid.

Compared to simpler random-graph models like the Erdős-Rényi or configuration models, the SBM has a large number of parameters. Adding things up, there are $1 + n + O(k^2)$ parameters, where second term is the group assignments and the last term counts the number of free parameters in the matrix M. In the undirected case, there are $\binom{k}{2}$ values in M that need to be specified before we can generate any edges, even if we have already chosen the labeling on the vertices. This flexibility allows the SBM to produce a wide variety of large-scale structures. Before discussing how to infer structure from data, we will explore some examples of how different choices of parameters produce different types of large-scale structure.

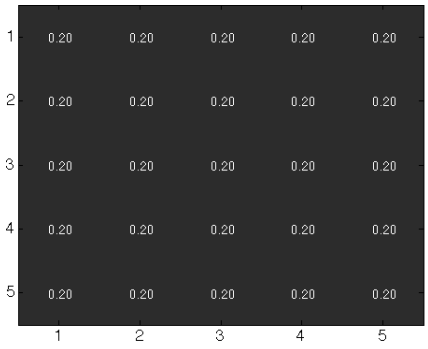
2.2.1 Random graphs

The SBM includes Erdős-Rényi random graph model $G(n, p)$ as a special case, and there are two ways to see this. First, we can specify $k = 1$, in which case there is only a single group and M reduces to a single parameter p. Now, the probability that two vertices i, j are connected is $M_{z(i),z(j)} = p$ constant because $z(i) = z(j)$ for every vertex. Alternatively, we can choose $k > 1$ and set $M_{u,v} = p$ constant for all pairs of groups u, v.

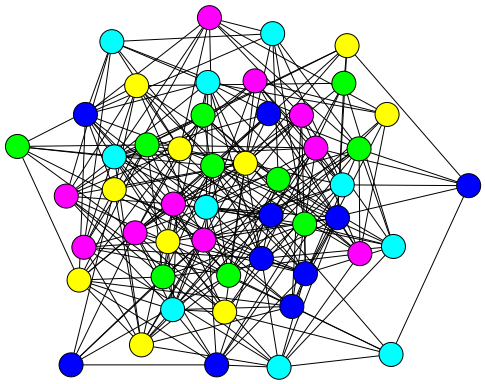
In each of these cases, all the mathematical results from simple random graphs apply, e.g., the degree distribution is a Poisson, the diameter is logarithmic, the expected clustering coefficient is vanishing, and a giant component exists for $\langle k \rangle > 1$. For $k = 5$, the figure on the next page shows an example of a stochastic block matrix M and a corresponding network instance drawn from it.

Although this point may appear trivial, it sheds some light on the general manner by which the SBM generates large-scale structural patterns. For instance, if $u = v$, then a single parameter

M_{uv} governs the probability of edges between vertices in that group. That is, the structure *within* groups in the SBM is a simple random graph. And, because a different single value specifies the density of edges for groups $u \neq v$, the structure *between* groups is a simple bipartite random graph. The key to the SBM's flexibility is that it can both choose how to assign vertices to groups and choose different densities for the induced random graphs.



random graph block matrix



random graph

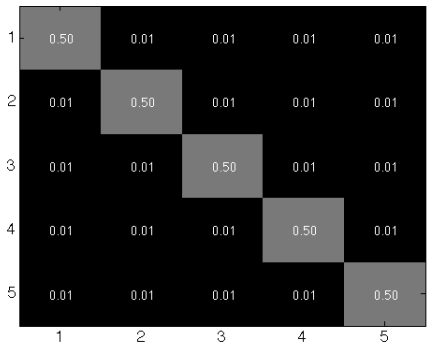
2.2.2 Assortative and disassortative communities

When communities are assortative, then vertices tend to connect to vertices that are like them, i.e., there are relatively more edges within communities. Under the SBM, assortative community structure appears as a pattern on M where the values on the diagonal are generally greater than the values off the diagonal. That is, $M_{uu} > M_{uv}$ for $u \neq v$. Similarly, disassortative structure implies that unlike vertices are more likely to connect than like vertices, i.e., $M_{uu} < M_{uv}$ for $u \neq v$.

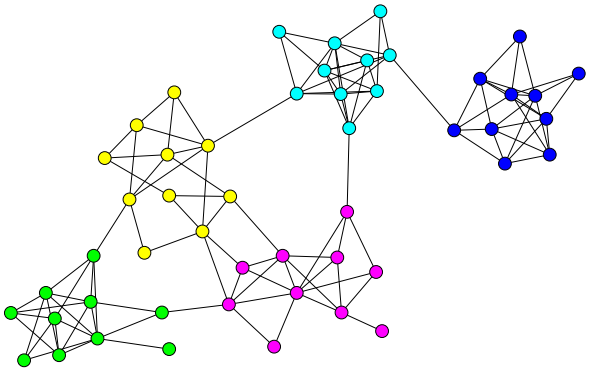
The figures on the next page illustrate these patterns, where each network has the same mean degree. Notably, the disassortative network looks visually similar to the ER network above, but this hides the fact that vertices with similar colors are not connecting with each other. In contrast, the assortative network shows nicely what we often expect for "community structure," and this pattern is what the modularity function Q prefers.

2.2.3 Core-periphery and ordered communities

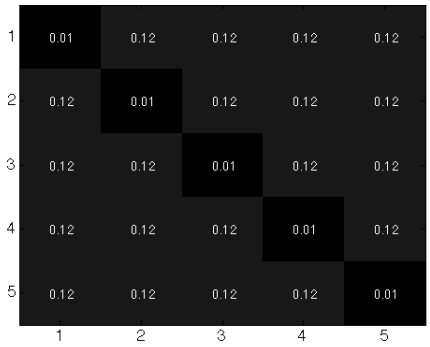
In an ordered network, groups connect to each other based on their position within some latent ordering.



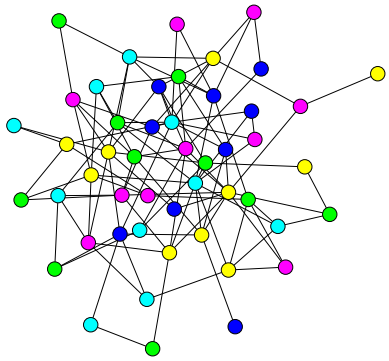
assortative block matrix



assortative communities



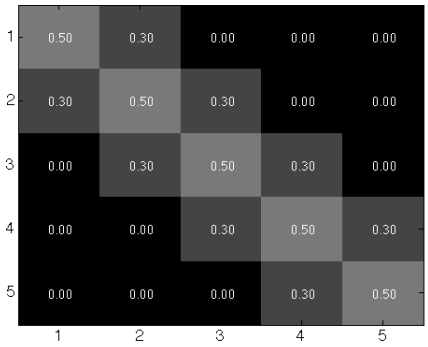
disassortative block matrix



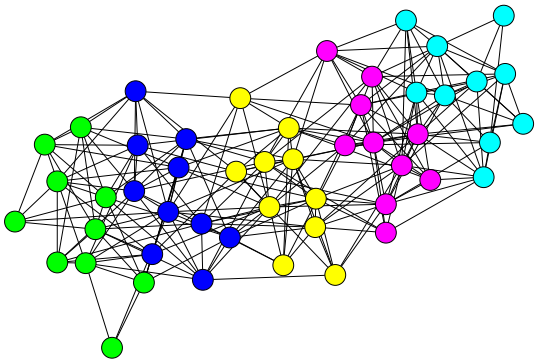
disassortative communities

For humans, physical proximity exhibits this kind of structure with age acting as a latent ordering variable. That is, individuals tend to associate with others who are close to themselves in age, so that children tend to associate preferentially with other children, teenagers with teenagers, 20-somethings with 20-somethings, etc. This induces a strong diagonal component in the block matrix, as in assortative communities, plus a strong first-off-diagonal component, i.e., communities connect to those just above and below themselves in the latent ordering $M_{uu} \approx M_{u,u+1} \approx M_{u,u-1}$. In social networks, an exception to this pattern occurs during the child-bearing years, so that individuals split their time between their peers and their children (who are generally 20-30 years younger).[3] The figure on the next page gives an example of this kind of block matrix and a single instance of a corresponding network drawn from this SBM.

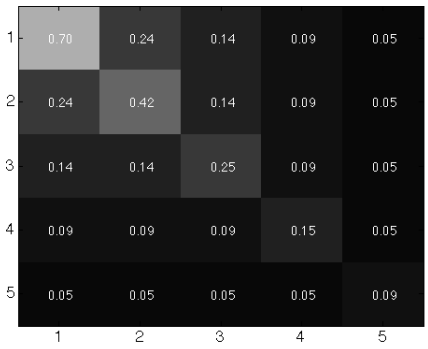
[3]This fact was demonstrated nicely using real empirical data on social association across multiple countries in Mossong et al., "Social Contacts and Mixing Patterns Relevant to the Spread of Infectious Diseases." *PLoS Medicine* **5**(3), e74 (2008).



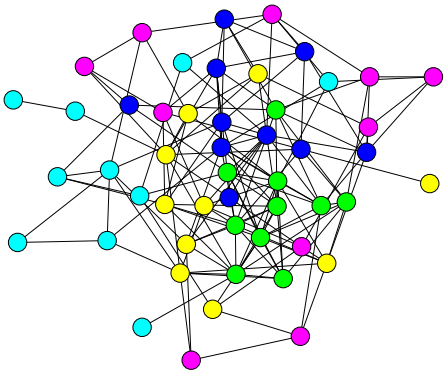
ordered block matrix



ordered communities



core-periphery block matrix



core-periphery network

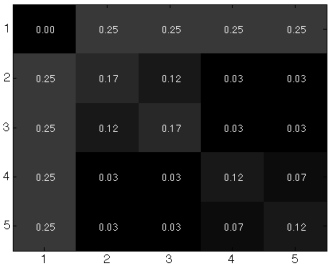
Core-periphery structure is a form of ordering on communities, but where we place the additional constraint that the density of connections decreases with the community index. The figure on the following page illustrates this pattern, where each "layer" connects to all other layers, but with exponentially decreasing probability. In the block matrix, one can see evidence in the upper left corner of the nested structure of this core-periphery network. In the network instance, the green vertices are the inner core, while the magenta and cyan vertices are the outer periphery.

2.2.4 Degree heterogeneity

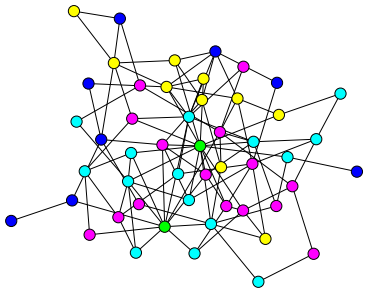
The networks that the SBM generates are composed of simple random graphs inside each group and simple random bipartite graphs between each pair of distinct groups. The degree distribution of the full network is thus always a mixture of Poisson degree distributions. Each bundle of edges

contributes to the degrees of the vertices it runs between, and so if its density is large, it will contribute many more edges to the degrees of its end points. We can use this flexibility to create more heavy-tailed degree distributions than we would normally expect from a simple random graph by placing a small number of vertices in a group with large densities to other, larger groups.

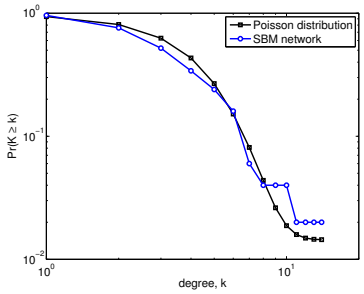
The following hand-constructed example illustrates this idea, where we choose the number of vertices in each group to be $\{2, 8, 10, 15, 15\}$. In the stochastic block matrix, the smallest group, with 2 vertices (green, in the network image), connects to 0.25 of the other vertices, and thus each of these vertices has expected degree $\mathrm{E}[k] = 12$, which is about twice as large as the expected degree of the other vertices. (Do you see how to calculate $\mathrm{E}[k]$?)



stochastic block matrix



heterogeneous degrees



degree distribution

2.2.5 Directed or undirected

As a final comment, the SBM can naturally handle directed networks, by relaxing the previous assumption that the stochastic block matrix be symmetric. In this way, the probability of an edge running from $i \to j$ can be different from the probability of an edge running in the opposite direction, from $j \to i$.

2.3 Fitting the SBM to data

Given a choice of k and an observed network G, we can use the SBM to infer the latent community assignments z and stochastic block matrix M. There are several ways of doing this, the simplest of which is to use maximum likelihood. That is, we aim to choose z and M such that we maximum the likelihood of generating exactly the edges observed in G. The likelihood of the data, given the

model is

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

where we have separated the terms corresponding to edges we observe $(i,j) \in E$ and those we do not $(i,j) \notin E$. Thus, every pair of vertices i, j appears in the likelihood function, and the function contains $O(n^2)$ terms.[4]

2.3.1 The maximum likelihood choice

In general, z and M can assume any values and the likelihood remains well defined. However, because we aim to maximize the probability of the SBM generating G, only one particular choice of M corresponds to this choice, which is the maximum likelihood choice of M conditioned on the partition z. This simplifies the inference considerably. (Choosing z, M can also be done using Bayesian approaches; we will not cover those here.)

Observe that each pair of group labels u, v identifies a "bundle" of edges, i.e., edges that run from group u to group v.[5] Under the SBM, each of these edges is iid with parameter M_{uv}, implying that the number of edges we actually observe in this bundle follows a Binomial distribution. For notational simplicity, let N_{uv} count the number of *possible* edges between groups u and v, and let N_u count the number of vertices with label u.

Regardless of whether we are modeling a network with self-loops or directed edges, when $u \neq v$, the number of possible edges is always $N_{uv} = N_u N_v$.

If our network is directed and can contain self-loops, then the number of possible within-group edges $N_{uu} = N_u^2$. If we prohibit self-loops, then it is $N_{uu} = N_u(N_u - 1)$. Finally, if the network is simple (no self-loops and undirected), then the number of possible edges is $N_{uu} = \binom{N_u}{2}$. In the equations below, we will use the more general term N_{uv}, but this should be replaced with the

[4] Whether it is n^2 or $n^2 - n$ or $\binom{n}{2}$ terms depends on whether we are modeling a directed network with self-loops, directed without self-loops, or a simple network. That is, if an edge cannot exist between a pair of vertices by definition of the network type, then that pair mustn't contribute to the total likelihood of the observed data. Furthermore, if the existence of an edge is already accounted for by some other pair of groups, e.g., v, u and u, v when the network is undirected, then it cannot contribute a second time to the likelihood.

[5] Note: the pair v, u only denotes a distinct edge bundle from that of u, v if the network is directed. If the network is undirected, the calculation only runs over the $\binom{k}{2}$ unique pairs of group labels.

appropriate expression when the model is applied to real data.

Suppose that for some particular vertex labeling z, and particular choice of groups u and v, we observe E_{uv} edges in the u, v edge bundle. Because this number is binomially distributed, the maximum likelihood choice for the probability M_{uv} that any particular edge in this bundle exists is simply the MLE for a binomial with expected value E_{uv}. Our estimate is thus $\hat{M}_{uv} = E_{uv}/N_{uv}$, which can easily be derived by counting the edges in a bundle, given the network G and a particular partition z.

We can now considerably simplify the likelihood function of Eq. (1). First we observe that because vertices are stochastically equivalent, we can change the product series to run over all pairs of groups, rather than over all pairs of vertices. Each pair of groups corresponds to a bundle of N_{uv} pairs of which we observe E_{uv} edges, each of which is iid with probability M_{uv}. Thus, we can write

$$\mathcal{L}(G \,|\, M, z) = \prod_{u,v} M_{uv}^{E_{uv}} (1 - M_{uv})^{N_{uv} - E_{uv}} \tag{2}$$

$$= \prod_{u,v} \left(\frac{E_{uv}}{N_{uv}}\right)^{E_{uv}} \left(1 - \frac{E_{uv}}{N_{uv}}\right)^{N_{uv} - E_{uv}} \quad , \tag{3}$$

where in the last step we have substituted the MLE for M_{uv} into the previous expression.

Taking the logarithm yields

$$\ln \mathcal{L} = \sum_{u,v} E_{uv} \ln \frac{E_{uv}}{N_{uv}} + (N_{uv} - E_{uv}) \ln \left(\frac{N_{uv} - E_{uv}}{N_{uv}}\right) \quad .$$

Now applying the rules of logarithms and collecting like terms yields, we obtain

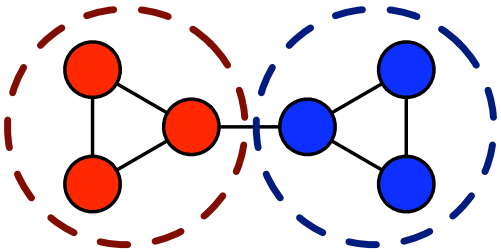
$$\begin{aligned}
\ln \mathcal{L} &= \sum_{u,v} E_{uv} \ln E_{uv} - E_{uv} \ln N_{uv} + (N_{uv} - E_{uv}) \left(\ln (N_{uv} - E_{uv}) - \ln N_{uv}\right) \\
&= \sum_{u,v} E_{uv} \ln E_{uv} - E_{uv} \ln N_{uv} + N_{uv} \ln (N_{uv} - E_{uv}) - N_{uv} \ln N_{uv} - E_{uv} \ln (N_{uv} - E_{uv}) + E_{uv} \ln N_{uv} \\
&= \sum_{u,v} E_{uv} \ln E_{uv} + N_{uv} \ln (N_{uv} - E_{uv}) - N_{uv} \ln N_{uv} - E_{uv} \ln (N_{uv} - E_{uv}) \\
&= \sum_{u,v} E_{uv} \ln E_{uv} + (N_{uv} - E_{uv}) \ln (N_{uv} - E_{uv}) - N_{uv} \ln N_{uv} \quad , \qquad (4)
\end{aligned}$$

which is a function that depends only on the counts induced by z.[6]

[6] Where we define $0^0 = 1$. This choice is necessary to prevent the numerical calculation from failing when either 0 edges run between, or 0 edges do not run between, a pair of groups.

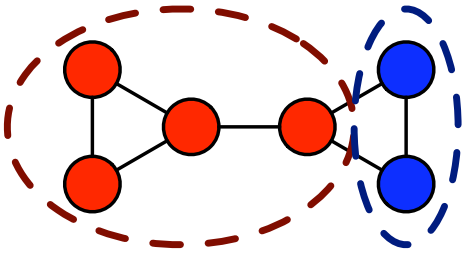
2.4 An example

Returning to the two-triangles network, we can tabulate the maximum likelihood stochastic block matrix M for each of the usual two partitions, and thus compute their likelihoods. To do so, we use the version of the SBM that generates simple networks, i.e., we restrict the summation to exclude self-loops and we count edges between groups only once. Applying Eq. (4) to the corresponding matrices shows that the "good" partition is about 177 times more likely to generate the observed data than the "bad" partition.



$\mathcal{L}_{\text{good}} = 0.043304\ldots$
$\ln \mathcal{L}_{\text{good}} = -3.1395\ldots$

M_{good}	red	blue
red	3/3	1/9
blue	1/9	3/3



$\mathcal{L}_{\text{bad}} = 0.000244\ldots$
$\ln \mathcal{L}_{\text{bad}} = -8.3178\ldots$

M_{bad}	red	blue
red	4/6	2/8
blue	2/8	1/1

Although the qualitative results are the same as for using the modularity function—the good partition is better—this likelihood-based approach provides additional information in the form of the likelihood ratio. That is, we now know just how much better, in probabilistic terms, the better partition is.

2.5 Choosing the number of groups k

Recall that we fixed k the number of groups. In many applications, we would like to allow k to vary and thus decide whether some choice $k' > k$ is better.

Because k determines the "size" of the model, allowing k to vary presents a difficulty: the larger a value of k we choose, the more parameters we have in M, which may lead to over fitting. In the limit of $k = n$, every vertex is in a group by itself, the matrix $M = A$ (the adjacency matrix), and the likelihood is maximized at $\mathcal{L} = 1$. That is, the model will *memorize* the data exactly. In other words, as we increase k, the distribution $\Pr(G \,|\, M, z)$ becomes increasingly concentrated around

the empirically observed network G until it places unit weight on G and no weight on any other graph.

The need to choose k for the SBM may appear to be a limitation relative to modularity maximization, which had no free parameter controlling its model complexity. This limitation is not as great as we might imagine, however, as the modularity function has a built in preference for modules with certain characteristics, which the SBM lacks; see Section 5 of the modularity lecture.

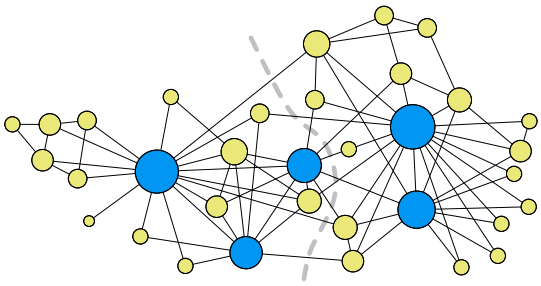
For the SBM, there are several statistically principled ways to choose k. Each of these require additional steps and each is based on slightly different assumptions about what makes a *good* model. Generally speaking, each of these methods computes some convex or concave function of the likelihood and the number of groups k, and chooses the best k as the maximum or minimum of this function. All of these methods are a kind of *regularization* or complexity control that effectively penalizes "larger" models for their additional flexibility. That is, we would only want to use a larger model (larger k) if the additional flexibility was statistically warranted. Popular choices for regularization include Bayesian marginalization, Bayes factors, various information criteria (BIC, AIC, etc.), minimum description length (MDL) approaches, and likelihood ratio tests. We will not cover any of these techniques here.[7]

2.6 Correcting for degree heterogeneity

In recent years, the SBM has become a popular model upon which to build more sophisticated generative models of community structure, with many variations.

One particularly nice variation is the so-called "degree corrected" SBM, by Karrer and Newman, which was motivated by the fact that when you apply the SBM to networks with skewed degree distributions, the model tends to group vertices by degree. For instance, consider the karate club network. Setting $k = 2$ yields the following labeling of the vertices, which does not correspond to the "true" or socially observed groups.
It is not hard to compute the stochastic block matrices corresponding to this division, which places the five highest-degree vertices in one group and all other vertices in the other group, and to the socially observed division. Given these, we can then compute the log-likelihood scores for the two partitions. The following tables show the results, indicating that, indeed, the SBM division is more likely (more positive log-likelihood), by a substantial margin. In fact, the SBM division is $\exp(198.50 - 179.39) \approx 10^8$ times more likely to generate the observed edges than the social division.

[7] At a future date, I will add references to some of these approaches here.



karate club, with SBM $k = 2$

M_{social}	A (17)	B (17)
A (17)	35/136	11/289
B (17)	11/289	32/136
A (17)	0.2574	0.0381
B (17)	0.0381	0.2353

social division, $\ln \mathcal{L} = -198.50$

M_{SBM}	A (5)	B (29)
A (5)	5/10	54/145
B (29)	54/145	19/406
A (5)	0.5000	0.3724
B (29)	0.3724	0.0468

SBM division, $\ln \mathcal{L} = -179.39$

Why does the SBM do this? Recall that the SBM can only decompose a network into combinations of random bipartite graphs and Erdős-Rényi random graphs, each of which has a Poisson degree distribution with mean $M_{uv}N_u$. Thus, if a network exhibits a more skewed degree distribution, as in the case of the karate club, the model correctly recognizes that in order to reproduce this pattern, it should place those high-degree vertices in a small group together with a large probability of connecting to other larger groups. In short, the likelihood function is maximized when each M_{uv} value is close to either 0 or 1, and the SBM thus prefers partitions that produce this kind of pattern. When a graph is sparse, a large but very weakly connected group is better, from the SBM's perspective, than two moderately sized but denser groups. Hence, the observed SBM division.

The downside of this tendency of the SBM is that a skewed degree distribution, like the one we see in the karate club, is a kind of violation of the SBM's particular assumption of edge independence, and the SBM seeks to explain it over other kinds of latent group structure that we might care about.

The degree-corrected SBM modifies the generative model in a way that allows vertices to have arbitrary degrees without having to force the combination of z and M to produce them. In addition to the usual SBM parameters, we add to each vertex a "propensity" parameter γ_i that controls the expected degree of vertex i. Recall that the model for the number of edges between a pair of vertices i and j in the SBM is a Bernoulli distribution. In the degree-corrected SBM, we simply replace this Bernoulli distribution with a Poisson distribution with mean $\gamma_i\gamma_j M_{z_i z_j}$.

The probability of observing the network G with adjacency matrix A is then

$$\begin{aligned} P(G \,|\, \gamma, M, z) &= \prod_{i,j} \text{Poisson}(\gamma_i \gamma_j M_{z_i z_j}) \\ &= \prod_{i<j} \frac{\left(\gamma_i \gamma_j M_{z_i z_j}\right)^{A_{ij}}}{A_{ij}!} \exp\left(-\gamma_i \gamma_j M_{z_i z_j}\right) \\ &\quad \times \prod_i \frac{\left(\frac{1}{2}\gamma_i^2 M_{z_i z_j}\right)^{A_{ii}/2}}{(A_{ii}/2)!} \exp\left(-\frac{1}{2}\gamma_i^2 M_{z_i z_j}\right) \ , \end{aligned} \tag{5}$$

where the composite likelihood appears because we assume an undirected network, and thus need to count edges within groups differently from edges between groups. When the Poisson mean $\gamma_i \gamma_j M_{z_i z_j}$ is typically very small, i.e., close to 0, we get a network that is sparse. This reformulation also implies that the networks produced are no longer simple, but are instead undirected multigraphs, as occasionally we will produce multiple edges between a pair i, j.

The propensity parameters in the Poisson mean are arbitrary to within a multiplicative constant, which we can absorb into the stochastic block matrix M. This observation allows us to normalize the propensity scores

$$\sum_i \gamma_i \delta_{u,z_i} = 1 \ , \tag{6}$$

for all group labels u, and where $\delta(u, v) = 1$ if $u = v$ and 0 otherwise. This constraint implies that γ_i is equal to the probability that an edge emerging from the group z_i will connect to vertex i, and allows us to simplify Eq. (5) as

$$P(G \,|\, \gamma, M, z) = C \prod_i \gamma_i^{k_i} \prod_{u,v} M_{uv}^{E_{uv}/2} \exp\left(-\frac{1}{2} M_{uv}\right) \ , \tag{7}$$

where k_i is the degree of vertex i, C is a constant (see below), and E_{uv} is the total number of edges between groups u and v or twice that number for $u = v$ (again, because we assume an undirected network),

$$E_{uv} = \sum_{ij} A_{ij} \delta_{u,z_i} \delta_{v,z_j} \ . \tag{8}$$

The constant C depends only on the adjacency matrix A, and includes the various factorials that come out of the Poisson distributions in Eq. (5)

$$C = \left(\prod_{i<j} A_{ij}! \prod_i 2^{A_{ii}/2} (A_{ii}/2)! \right)^{-1} \ . \tag{9}$$

Taking derivatives of the log-likelihood function (derived from Eq. (7)) allows us to write down the maximum likelihood estimators[8] for the model parameters, given a partition z. These have a particular nice form:

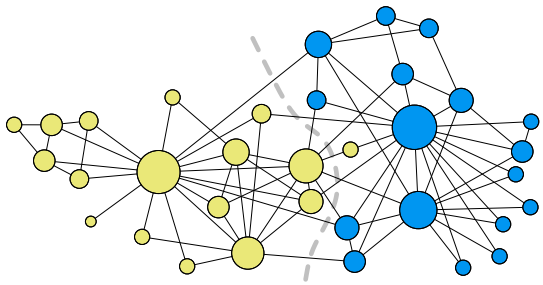
$$\hat{\gamma}_i = \frac{k_i}{\kappa_{z_i}} \qquad\qquad \hat{M_{uv}} = E_{uv} \ , \tag{10}$$

where κ_u is the sum of the degrees in group u, i.e., the total degree of the community. As with the SBM, we can further simplify the form of the likelihood function by substituting these MLEs into its form. The result is a fairly compact expression that again depends only on the counts induced by the choice of partition z:

$$\ln \mathcal{L}(G \,|\, z) = \sum_{uv} \frac{E_{uv}}{2m} \ln \frac{E_{uv}/2m}{(\kappa_u/2m)(\kappa_v/2m)} \ . \tag{11}$$

Notably, this form is similar in some ways to the modularity function, which includes terms for the expected number of edges between a pair of groups, conditioned on the fraction of all edges attached to those groups. Thus, the degree corrected SBM (or DC-SBM, if you like acronyms) can be thought of as seeking a partition that maximizes the "information" contained in the labels relative to a random graph with a given degree sequence, while the SBM seeks the same relative to a different null model, the Erdős-Rényi random graph.

Applying the DC-SBM to the karate club allows the propensity parameters to generate more skewed degrees within communities, and yields an inferred division that is much closer to the truth.[9]



with degree correction

[8] Try this at home.

[9] There is a fun story associated with the one misclassified vertex, which has an equal number of connections to each of the two groups. In the original club, this person apparently had a karate exam coming up soon, and when the club split in two, they chose to go with the instructor's faction instead of the president's in order to be better prepared for the exam.

3 Taking stock of our random graph models

The stochastic block model is another kind of random graph model, which means we can use it as a null model, for deciding whether some empirically observed pattern can be explained by group structure alone (since that is what the SBM reproduces) or group+degree structure (via the degree-corrected SBM). And, because we can estimate its parameters from real data, we can use the SBM as the basis for machine-learing-type approaches to analyzing and modeling network structure.

Below is an updated version of our Table summarizing and comparing the properties of our random graph models. Notably, for the SBM, many of its properties now depend on the mixing matrix M: if M contains assortative community structure, then the clustering coefficient and reciprocity (in a directed version) can be much higher than in a typical random graph model, while if M contains disassortative communities, then it will be low, like other random-graph models.

network property	real-world	Erdős-Rényi	configuration	SBM	DC-SBM
degree distribution	heavy tailed	Poisson($\langle k \rangle$)	specified	Poisson mixture	specified
diameter	"small" ($\propto \log n$)	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
clustering coefficient	social: moderate non-social: low	$O(1/n)$	$O(1/n)$	depends on M	depends on M
reciprocity	high	$O(1/n)$	$O(1/n)$	depends on M	depends on M
giant component	very common	$\langle k \rangle > 1$	$\langle k^2 \rangle - 2\langle k \rangle > 0$	depends on M	depends on M

4 At home

1. Read Chapter 8 (pages 359–418) in *Pattern Recognition*
2. Next time: fitting block models to data