Oriented and Degree-generated Block Models: Generating and Inferring Communities with Inhomogeneous Degree Distributions

Yaojia Zhu
Computer Science
University of New Mexico
yaojia.zhu@gmail.com

Xiaoran Yan
Computer Science
University of New Mexico
everyan@cs.unm.edu

Cristopher Moore
Santa Fe Institute
and University of New Mexico
moore@santafe.edu

Abstract

The stochastic block model is a powerful tool for inferring community structure from network topology. However, it predicts a Poisson degree distribution within each community, while most real-world networks have a heavy-tailed degree distribution. The degree-corrected block model can accommodate arbitrary degree distributions within communities. But since it takes the vertex degrees as parameters rather than generating them, it cannot use them to help it classify the vertices, and its natural generalization to directed graphs cannot even use the orientations of the edges. In this paper, we present variants of the block model with the best of both worlds: they can use vertex degrees and edge orientations in the classification process, while tolerating heavy-tailed degree distributions within communities. We show that for some networks, including synthetic networks and networks of word adjacencies in English text, these new block models achieve a higher accuracy than either standard or degree-corrected block models.

1 Introduction

In many real-world networks, vertices can be divided into *communities* based on their connections. Social networks can be forged by daily interactions like karate training [21]. The blogosphere contains groups of linked blogs with similar political views [1]. Words can be tagged as different parts of speech based on their adjacencies in large texts [14]. Communities range from assortative clumps, where vertices preferentially attach to others of the same type, to *functional* communities of vertices that connect to the rest of the network in similar ways, such as groups of predators in a food web that feed on similar prey [3, 12]. Understanding various community structures, and their relations to the functional roles of vertices and edges, is crucial to understanding network data.

The *stochastic block model* (SBM) [8, 10, 19, 2] is a popular and highly flexible generative model for community detection. It partitions the vertices into communities or *blocks*, where vertices belonging to the same block are *stochastically equivalent* [20] in the sense that the probabilities of a connection with all other vertices are the same for all vertices in the same block. With the general definition of community, block models can capture many types of community structure, including assortative, disassortative, and satellite communities and mixtures of them [15, 16, 13, 12, 7, 6].

The SBM assumes that each edge is generated independently conditioned on the block memberships. Each entry A_{uv} of the adjacency matrix is then Bernoulli-distributed, where the probability that $A_{uv} = 1$ depends solely on the block memberships g_u, g_v of its endpoints. Since every pair of vertices in a given pair of blocks are connected with the same probability, for large n the degree distribution within each block is Poisson. As a consequence, vertices with very different degrees are unlikely to be in the same block. This leads to problems when modeling real networks, which often have heavy-tailed degree distributions within each community. For instance, both liberal and conservative political blogs range from high-degree "leaders" to low-degree "followers" [1].

To avoid this effect, and allow degree inhomogeneity within blocks, there is a long history of generative models where the probability of an edge depends on node attributes θ_u as well as their group memberships (e.g. [13, 18]). Karrer and Newman [11] introduced the *degree-corrected* (DC) block model. They consider random multigraphs, where A_{uv} is Poisson-distributed with mean $\theta_u\, \theta_v\, \omega_{g_u,g_v}$. The most-likely value of θ_u is the degree d_u, and this model can thus generate graphs with arbitrary (expected) degree distributions within each community.

On the other hand, the degree-corrected model cannot use the vertex degrees to help it classify the vertices, precisely because it takes the degrees as parameters rather than as data that need to be explained. For this reason, DC may actually fail to recognize communities that differ significantly in their degree distributions. Thus we have two extremes: the SBM separates vertices by degree even when it shouldn't, and DC fails to do so even when it should. For directed graphs, the natural generalization of DC, the *directed degree-corrected* (DDC) block model, has two parameters for each vertex: the expected in-degree and out-degree. But this model cannot even take advantage of edge orientations. For instance, in English adjectives usually precede nouns but rarely vice versa. Thus the ratio of each vertex's in- and out-degree is strongly indicative of its block membership, and leveraging this part of the data is very helpful for classification.

In this paper, we propose two new types of block model, which combine the strengths of the degree-corrected and uncorrected block models. The *oriented degree-corrected* (ODC) block model is able to utilize the edge orientations for community detection by only correcting the total degrees. We show that for networks with strongly asymmetric behavior between communities, including synthetic networks and networks of word adjacencies in English text, ODC achieves a higher accuracy.

We also propose the *degree-generated* (DG) block model, which treats the expected degree of each vertex as generated from prior distributions, such as power laws whose exponents vary from one community to another. By including the probability of these degrees in the likelihood of a given block assignment, the model captures the interaction between the degree distribution and the community structure. DG automatically strikes a balance between allowing vertices of different degrees to coexist in the same community on the one hand, and using vertex degrees to separate vertices into communities on the other. Our experiments show that DG works especially well in networks where communities have highly inhomogeneous degree distributions, but where the degree distributions differ enough between communities so that we can use vertex degrees to help us classify the vertices. In some cases, DG has a further advantage in faster convergence as it reshapes the parameter space, providing the algorithm a shortcut to the correct community structure.

These new variants of the block model give us the best of both worlds. They can tolerate heavy-tailed degree distributions within communities, but can also use degrees and edge orientations to help classify the vertices. In addition to their performance on these networks, our models illustrate a valuable point about generative models and statistical inference: when inferring the structure of a network, you can only use the information that you try to generate.

2 The models

2.1 Background: degree-corrected block models

Throughout, we use N and M to denote the number of vertices and edges, and K to denote the number of blocks. The problem of determining the number of blocks is a subtle model selection problem, which we do not address here.

In the original stochastic block model, the entries A_{uv} of the adjacency matrix are independent and Bernoulli-distributed, with $P(A_{uv} = 1) = p_{g_u,g_v}$. Here g_u is the block to which u belongs, where p is a $K \times K$ matrix. Karrer and Newman [11] consider random multigraphs where the A_{uv} are independent and Poisson-distributed, $A_{uv} \sim \mathrm{Poi}(\theta_u \theta_v \omega_{g_u,g_v})$. Here ω replaces p, and θ_u is an overall propensity for u to connect to other vertices. Note that since the A_{uv} are independent, the degrees d_u will vary somewhat around their expectations; however, the resulting model is much simpler to analyze than one that controls the degree of each vertex exactly.

Ignoring self-loops, the likelihood with which this degree-corrected (DC) block model generates an undirected multigraph G is then

$$P(G \mid \theta, \omega, g) = \prod_{u<v} \frac{(\theta_u \theta_v \omega_{g_u g_v})^{A_{uv}}}{A_{uv}!} \exp\left(-\theta_u \theta_v \omega_{g_u g_v}\right) . \tag{1}$$

To remove the obvious symmetry where we multiply the θ's by a constant C and divide ω by C^2, we can impose a normalization constraint $\sum_{u:g_u=r} \theta_u = \kappa_r$ for each block r, where $\kappa_r = \sum_{u:g_u=r} d_u$ is the total degree of the vertices in block r. Under these constraints, the maximum likelihood estimates (MLEs) for the θ parameters are $\hat{\theta}_u = d_u$. For each pair of blocks r, s, the MLE for ω_{rs} is

$$\hat{\omega}_{rs} = \frac{m_{rs}}{\kappa_r \kappa_s} ,$$

where m_{rs} is the number of edges connecting block r to block s (and edges within blocks are counted twice). Substituting these MLEs for θ and ω then gives the log-likelihood

$$\log P(G \mid g) = \frac{1}{2} \sum_{r,s=1}^{K} m_{rs} \log \frac{m_{rs}}{\kappa_r \kappa_s} . \tag{2}$$

2.2 Directed and oriented degree-corrected models

The natural extension of DC to directed networks, which we call the directed degree-corrected block model (DDC), has two parameters $\theta_u^{\text{out}}, \theta_u^{\text{in}}$ for each vertex. The number of directed edges from u to v is again Poisson-distributed, $A_{uv} \sim \text{Poi}(\theta_u^{\text{out}} \theta_v^{\text{in}} \omega_{g_u, g_v})$. We impose the constraints $\sum_{u:g_u=r} \theta_u^{\text{out}} = \kappa_r^{\text{out}}$ and $\sum_{u:g_u=r} \theta_u^{\text{in}} = \kappa_r^{\text{in}}$ for each block r, where $\kappa_r^{\text{out}} = \sum_{u:g_u=r} d_u^{\text{out}}$ and $\kappa_r^{\text{in}} = \sum_{u:g_u=r} d_u^{\text{in}}$ denote the total out- and in-degree of block r. As before, let m_{rs} denote the number of directed edges from block r to block s. Then the likelihood is

$$\begin{aligned} P(G \mid \theta, \omega, g) &= \prod_{uv} \frac{\left(\theta_u^{\text{out}}\, \theta_v^{\text{in}}\, \omega_{g_u g_v}\right)^{A_{uv}}}{A_{uv}!} \exp(-\theta_u^{\text{out}}\, \theta_v^{\text{in}}\, \omega_{g_u g_v}) \\ &= \frac{\prod_u (\theta_u^{\text{out}})^{d_u^{\text{out}}} (\theta_u^{\text{in}})^{d_u^{\text{in}}} \prod_{rs} \omega_{rs}^{m_{rs}} \exp(-\kappa_r^{\text{out}} \kappa_s^{\text{in}} \omega_{rs})}{\prod_{uv} A_{uv}!} , \end{aligned} \tag{3}$$

Ignoring constants, we get the log-likelihood as follows

$$\log P(G \mid \theta, \omega, g) = \sum_u (d_u^{\text{out}} \log \theta_u^{\text{out}} + d_u^{\text{in}} \log \theta_u^{\text{in}}) + \sum_{rs} (m_{rs} \log \omega_{rs} - \kappa_r^{\text{out}} \kappa_s^{\text{in}} \omega_{rs}) . \tag{4}$$

The MLEs for the parameters (see full paper in arXiv) are

$$\hat{\theta}_u^{\text{out}} = d_u^{\text{out}} , \quad \hat{\theta}_u^{\text{in}} = d_u^{\text{in}} , \quad \hat{\omega}_{rs} = \frac{m_{rs}}{\kappa_r^{\text{out}} \kappa_s^{\text{in}}} . \tag{5}$$

Substituting these MLEs gives

$$\log P(G \mid g) = \sum_{r,s=1}^{K} m_{rs} \log \frac{m_{rs}}{\kappa_r^{\text{out}} \kappa_s^{\text{in}}} . \tag{6}$$

In the DDC, the expected in- and out-degrees of each vertex are completely specified by the θ parameters. Thus the DDC allows vertices with arbitrary degrees to fit comfortably together in the same block. On the other hand, since the degrees are given as parameters, rather than as data that the model must generate and explain, the DDC cannot use them to infer node labels. Indeed, it cannot even take advantage of the orientations of the edges, as shown by its poor performance on networks with strongly asymmetric community structure.

To deal with this, we present a partially degree-corrected block model capable of taking advantage of edge orientations, which we call the *oriented degree-corrected* (ODC) block model. Following the maxim that we can only use the information that we try to generate, we correct only for the total degrees of the vertices, and generate the edges' orientations.

Let $\bar{G}$ denote the undirected version of a directed graph G, i.e., the multigraph resulting from erasing the arrows for each edge. Its adjacency matrix is $\bar{A}_{uv} = A_{uv} + A_{vu}$, so (for instance) $\bar{G}$ has two edges between u and v if G had one pointing in each direction. The ODC can be thought of as generating $\bar{G}$ according to the undirected degree-corrected model, and then choosing the orientation of each edge according to another matrix ρ_{rs}, where an edge (u, v) is oriented from u to v with probability ρ_{g_u,g_v}. Thus the total log-likelihood is

$$\log P(G \,|\, \theta, \omega, \rho, g) = \log P(\bar{G} \,|\, \theta, \omega, g) + \log P(G \,|\, \bar{G}, \rho, g) . \tag{7}$$

Writing $\bar{m}_{rs} = m_{rs} + m_{sr}$ and $\kappa_r = \kappa_r^{\mathrm{in}} + \kappa_r^{\mathrm{out}}$, we can set θ_u and ω_{rs} for the undirected model to their MLEs as in Section 2.1, giving

$$\log P(\bar{G} \,|\, g) = \frac{1}{2} \sum_{r,s=1}^{K} \bar{m}_{rs} \log \frac{\bar{m}_{rs}}{\kappa_r \kappa_s} . \tag{8}$$

The orientation term is

$$\log P(G \,|\, \bar{G}, \rho, g) = \sum_{rs} m_{rs} \log \rho_{rs} = \frac{1}{2} \sum_{rs} (m_{rs} \log \rho_{rs} + m_{sr} \log \rho_{sr}) , \tag{9}$$

For each r, s we have $\rho_{rs} + \rho_{sr} = 1$, and the MLEs for ρ are

$$\hat{\rho}_{rs} = m_{rs} / \bar{m}_{rs} . \tag{10}$$

As (9) is maximized when the $\hat{\rho}_{rs}$ are near 0 or 1, the edge orientation term prefers highly directed inter-block connections. Since $\hat{\rho}_{rr} = 1/2$ for any r, it also prefers disassortative mixing, with as few connections as possible within blocks. Substituting the MLEs for ρ and combining (8) with (9),

$$\log P(G \,|\, g) = \sum_{r,s=1}^{K} m_{rs} \log \frac{m_{rs}}{\kappa_r \kappa_s} . \tag{11}$$

We can also view the ODC as a special case of the DDC, where we add the constraint $\theta_u^{\mathrm{in}} = \theta_u^{\mathrm{out}}$ for all vertex u (see full paper in arXiv). Moreover, if we set $\theta_u = 1$ for all u, we obtain the original block model, or rather its Poisson multigraph version where each A_{uv} is Poisson-distributed with mean ω_{g_u,g_v}. Thus SBM $\leq$ ODC $\leq$ DDC , where $A \leq B$ means that model A is a special case of model B, or that B is an elaboration of A. We will see below that since it is forced to explain edge orientations, the ODC performs better on some networks than either the simple SBM or the DDC.

2.3 Degree-generated block models

Another way to utilize vertex degrees for community detection is to require the model to generate them, according to some degree distribution derived from domain knowledge. For instance, many real-world networks have a power-law degree distribution, but with parameters (such as the exponent, minimum degree, or leading constant) that vary from community to community. In that case, the degree of a vertex gives us a clue as to its block membership. This leads to the *degree-generated* (DG) block models. They can tolerate heavy-tailed degree distributions within communities, but can also use degrees and edge orientations to help classify the vertices.

We generate the θ parameters of one of the degree-corrected block models discussed above, i.e., the expected vertex degrees, and use them to generate a random multigraph. Specifically, each θ_u is generated independently according to some distribution whose parameters ψ depend on the block g_u to which u belongs. Thus DG is a hierarchical model, which extends the previous degree-corrected block models by adding a degree generation stage on top, treating the θs as generated by the block assignment g and the parameters ψ rather than as parameters.

We can apply this approach to the undirected, directed, or oriented versions of the degree-corrected model; at the risk of drowning the reader in acronyms, we denote these DG-DC, DG-DDC, and DG-ODC. In each case, the total log-likelihood of a graph G is

$$\log P(G \,|\, \psi, \omega, g) = \log \int \mathrm{d}\theta \, P(G \,|\, \theta, \omega, g) \, P(\theta \,|\, \psi, g) ,$$

where

$$P(\theta \,|\, \psi, g) = \prod_u P(\theta_u \,|\, \psi_{g_u})\,.$$

For the directed models, we use θ_u as a shorthand here for θ_u^{in} and θ_u^{out}.

As in many hierarchical models, computing this integral appears to be difficult, except when $P(\theta \,|\, \psi)$ has the form of a conjugate prior such as the Gamma distribution (see full paper in arXiv). We approximate it by assuming that it is dominated by the most-likely value of θ,

$$\log P(G \,|\, \psi, \omega, g) \approx \log P(G \,|\, \hat{\theta}, \omega, g) + \log P(\hat{\theta} \,|\, \psi, g)\,.$$

However, even determining $\hat{\theta}$ is challenging when $P(\theta \,|\, \psi)$ is, say, a power law with a minimum-degree cutoff. Thus we make a further approximation, setting $\hat{\theta}$ just by maximizing the block model term $\log P(G \,|\, \hat{\theta}, \omega, g)$ as we did before, using (5) or the analogous equations for the DC or ODC. In essence, these approximations treat $P(\hat{\theta} \,|\, \psi, g)$ as a penalty term, imposing a prior on the degree distribution of each community with hyperparameters ψ. This leads to community structures that might not be as good a fit to the edges, but compensate with a much better fit to the degrees.

We can either treat the degree-generating parameters ψ as fixed (say, as predicted by a theoretical model of network growth [**?**, **?**, **?**]) or infer them by finding the $\hat{\psi}$ that maximizes $P(\hat{\theta} \,|\, \psi)$. For instance, suppose the θ_u in block $g_u = r$ are distributed as a continuous power law with a lower cutoff $\theta_{\min,r}$. Specifically, let the parameters in each block r be $\psi_r = (\alpha_r, \beta_r, \theta_{\min,r})$, and

$$P(\theta_u \,|\, \psi_r) = \begin{cases} \beta_r & \theta_u = 0 \\ 0 & 0 < \theta_u < \theta_{\min,r} \\ \frac{(1-\beta_r)(\alpha-1)}{\theta_{\min,r}} \left(\frac{\theta_u}{\theta_{\min,r}}\right)^{-\alpha_r} & \theta_u \geq \theta_{\min,r}\,. \end{cases}$$

In the directed case, we have $\psi_r^{\text{in}} = (\alpha_r^{\text{in}}, \beta_r^{\text{in}}, \theta_{\min,r}^{\text{in}})$ and $\psi_r^{\text{out}} = (\alpha_r^{\text{out}}, \beta_r^{\text{out}}, \theta_{\min,r}^{\text{out}})$. Allowing β_r^{out} to be nonzero, for instance, lets us directly include nodes with no outgoing neighbors; we find this useful in some networks. Alternately, we can choose $(\theta_u^{\text{in}}, \theta_u^{\text{out}})$ from some joint distribution, allowing in- and out-degrees to be correlated in various ways.

We fix $\theta_{\min,r} = 1$. Given the degrees and the block assignment, let $Y_r = \{u : g_u = r \text{ and } \theta_u \neq 0\}$, and let $y_r = |Y_r|$. The MLE for α_r is [4]

$$\hat{\alpha}_r = 1 + y_r \Bigg/ \sum_{u \in Y_r} \ln \theta_i\ . \tag{12}$$

The MLE for $\hat{\beta}_r$ is simply the fraction of vertices in block r with degree zero.

3 Experimental results

3.1 Experiments on synthetic networks

In order to understand under what circumstances our models out-perform previous variants of the block model, we performed experiments on synthetic networks, varying the degree distributions in communities, the degree of directedness between communities, and so on. First, we generated undirected networks according to the DG-DC model, with two blocks of equal size $N/2$. In order to confound the block model as much as possible, we deliberately designed these networks so that the two blocks have the same average degree. The degree distribution in block 1 is a power law with exponent $\alpha = 1.7$, with an upper bound of 1850, so that the average degree is 20. The degree distribution in block 2 it is Poisson, also with mean 20. As shown in the full paper in arXiv, the upper bound on the power law is larger than any degree actually appearing in the network; it really just changes the normalizing constant of the power law, and the MLE for α can still be calculated using (12). We assume the algorithm knows that one block has a power law degree distribution and the other is Poisson, but we force it to infer the parameters of these distributions.

As in [11], we use a parameter λ to interpolate linearly between a fully random network with no community structure and a "planted" one where the communities are completely separated. Thus

$$\omega_{rs} = \lambda \omega_{rs}^{\text{planted}} + (1-\lambda)\omega_{rs}^{\text{random}}$$

where

$$\omega_{rs}^{\text{random}} = \frac{\kappa_r \kappa_s}{2M}\,,\ \omega^{\text{planted}} = \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix}.$$

We inferred the community structure with various models. We ran the Kernighan-Lin (KL) heuristic first to find a local optimum [11], and then ran the heat-bath MCMC algorithm with fixed number of iterations to further refine it if possible. We initialized each run with a random block assignment; to test its stability, we also tried initializing them with the correct block assignment. Since isolated vertices don't participate in the community structure, giving us little basis on which we can classify them, we remove them and focus on the giant component. For $\lambda = 1$, where the community structure is purely the "planted" one, we kept two giant components, one in each community.

We measured accuracy by the normalized mutual information (NMI) [5] between the most-likely block assignment found by the model and the correct assignment. For groups of unequal size, the NMI is a better measure of accuracy than the fraction of vertices labeled correctly, since one can make this fraction fairly large simply by assigning every vertex to the larger group.

As shown in Fig. 1, DG-DC works very well even for small λ. This is because it can classify most of the vertices simply based on their degrees. As λ increases, it uses the connections between communities as well, giving near-perfect accuracy for $\lambda \geq 0.6$. It does equally well whether its initial assignment is correct or random. The DC model, in contrast, is unable to use the vertex degrees, and has accuracy near zero (i.e., not much better than a random block assignment) for $\lambda \leq 0.2$. Like the SBM [6, 7], it may have a phase transition at a critical value of λ below which the community structure is undetectable. Initializing it with the correct assignment helps somewhat at these values of λ, but even then it settles on an assignment far from the correct one. The original stochastic block model (SBM), which doesn't correct the degrees, separates vertices with high degrees from vertices with low degrees. Thus it cannot find the correct group structure even for large λ. Our synthetic tests are designed to have a broad degree distribution in block 1, and thus make SBM fail.

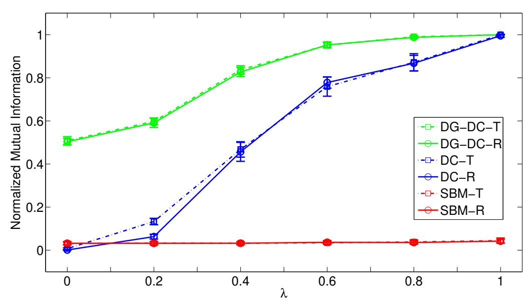

Figure 1: Tests on synthetic networks generated by the DG-DC model. Each point is based on 30 randomly generated networks with $N = 2400$. For each network and each model, we choose the best result from 10 independent runs, initialized either with random assignments (the suffix R) or the true block assignment (the suffix T). Each run consisted of the KL-heuristic followed by 10^6 MCMC steps. Our degree-generated (DG) block model performs much better on these networks than the degree-corrected (DC) model. The non-degree-corrected (SBM) model doesn't work at all.

We also did synthetic tests on directed graphs, with results similar to the real networks in the following section. For details please refer to the full paper in the arXiv.

3.2 Experiments on real networks

We studied three word adjacency networks, where vertices are separated into two blocks: adjectives and nouns. The first consists of common words in Dickens' novel *David Copperfield* [17]. The other two are built from the Brown corpus, which is a tagged corpus of present-day edited American English across various categories, including news, novels, documents, and many others [9]. The smaller one contains words in the News category (45 archives) that appeared at least 10 times; the larger one contains all the adjectives and nouns in the giant component of the entire corpus.

Network	#words	#adjective	#noun	#edges (S)	#edges (M)
David	112	57	55	569	1494
News	376	91	285	1389	2411
Brown	23258	6235	17023	66734	88930

Table 1: Basic statistics of the three word adjacency networks. S and M denote the simple and multigraph versions respectively.

	David(S)	David(M)	News(S)	News(M)	Brown(S)	Brown(M)
SBM	.423	.051	.006	.021	.001	7e-04
DC	**.566**	**.568**	.084	.015	.160	.155
ODC	.462	.470	**.247**	**.270**	**.311**	**.318**
DDC	.015	.060	.084	.005	.005	.070
NH	.395	.449	.215	.233	.309	.314

Table 2: Results using the naive NH assignment as the initial condition, again followed by 10^6 MCMC steps. ODC generally outperforms the other models.

We considered both the simple version of these networks where $A_{uv} = 1$ if u and v ever occur together in that order, and the multigraph version where $A_{uv} \geq 0$ is the number of times they occur together. The sizes, block sizes, and number of edges of these networks are shown in Table 1. In "News" and "Brown", the block sizes are quite different, with more nouns than adjectives. As discussed above, the NMI is a better measure of accuracy than the fraction of vertices labeled correctly.

In each network, both blocks have heavy-tailed in- and out-degree distributions (Fig. 2). The connections between them are disassortative and highly asymmetric: since in English adjectives precede nouns more often than they follow them, and more often than adjectives precede adjectives or nouns precede nouns, ω_{12} is roughly 10 times larger than ω_{21}, and ω_{12} is larger than either ω_{11} or ω_{22}.

Table 2 compares the performance of non-degree-generated block models, including SBM, DC, ODC, and DDC. (Under DC, we ignore the edge orientations, and treat the graph as undirected. Note that the resulting network may contain multi-edges even though the directed one doesn't). In our experiments, we started with a initial block assignment given by a naive heuristic (NH) which simply labels a vertex v as an adjective if $d_v^{\text{out}} > d_v^{\text{in}}$, and a noun if $d_v^{\text{in}} > d_v^{\text{out}}$ (If $d_v^{\text{out}} = d_v^{\text{in}}$, NH labels v randomly with equal probabilities). Then we ran the Kernighan-Lin (KL) heuristic to find a local optimum [11], and then ran the heat-bath MCMC algorithm.

For "David", DC and ODC work fairly well, and both are better than the naive NH. The standard SBM works well on "David(S)" but fails on "David(M)" because the degrees in the multigraph are more skewed than those in the simple one. Finally, DDC performs the worst; by correcting for in- and out-degrees separately, it loses any information that the edge orientations could provide. For "News" and "Brown", all these models fail except ODC, although it does only slightly better than the naive NH. Note that this more accurate assignment actually has lower likelihood than the one found using a random initial condition. NH initializes ODC into a more accurate, but less likely, local optimum, which other models fails even to capture.

Next, we shall test the performance of degree-generated models on the Brown network. According to Fig. 2, the in- and out-degree distributions in each block have heavy tails close to a power-law. Moreover, the out-degrees of the adjectives have a heavier tail than those of the nouns, and vice versa for the in-degrees. This is exactly the kind of difference in the degree distributions between communities that our DG block models are designed to take advantage of.

As Table 3 shows, degree generation improves DC and DDC significantly, letting them find a good assignment as opposed to one with NMI near zero. For ODC, the slight performance improvement makes DG-ODC the best model overall. We compare performance starting with the KL heuristic to performance using MCMC alone. We see that degree generation gives ODC almost as much benefit as the KL heuristic does. In other words, it speeds up the MCMC optimization process, letting ODC find a good assignment without the initial help of the computationally expensive KL heuristic.

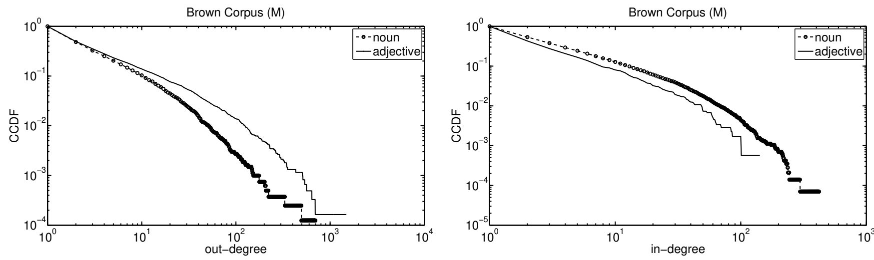

Figure 2: Degree distributions in the Brown network.

		Brown(S)			Brown(M)		
		DC	ODC	DDC	DC	ODC	DDC
–	–	.010	.188	.008	.007	.203	.011
KL	–	.020	.311	.016	.015	.318	.012
–	DG	.267	.302	.213	.278	.310	.149
KL	DG	.271	**.312**	.225	.284	**.320**	.195

Table 3: Performance of degree-generated models. KL indicates that we applied the KL heuristic before 10^6 MCMC steps. DG indicates degree generation. Each number gives the NMI for the most-likely assignment found in 50 independent runs. The best model is DG-ODC. Moreover, degree generation helps ODC converge, providing much of the benefit of the KL heuristic while avoiding its long running time (see bold numbers).

4 Conclusions

Degree correction in stochastic block models provides a powerful approach to dealing with networks with inhomogeneous degree distributions. However, in a sense it denies information to the inference process, since a generative model can only help us learn from the data that it has to generate.

We have introduced two new kinds of block models that allow for broad or heavy-tailed degree distributions, while using the degrees to help us detect communities. Unlike the directed degree-corrected (DDC) block model, which takes both in- and out-degrees as parameters, ODC is able to capture certain correlations between the in- and out-degrees. Simply put, for ODC, two vertices are unlikely to be in the same community if one has high in-degree and low out-degree while another has high out-degree and low in-degree. If the network is highly directed or asymmetric, the edge orientations can help ODC find community structures that DDC fails to perceive.

Our DG models use degree-corrected block models as a subroutine, but impose a penalty term based on the prior likelihood of the degree distribution in each community. DG models achieve high accuracy even when the density of connections between communities is close to uniform, as we illustrated in synthetic networks for small λ. Augmenting block models, such as the ODC, with degree generation also appears to speed up their convergence in some cases, helping simple algorithms like MCMC handle large networks without the benefit of expensive preprocessing steps like the KL heuristic. However, the effectiveness of DG depends heavily on knowing the correct form of the degree distribution in each community.

With all these variants of the block model, ranging from the "classic" version to degree-corrected and degree-generated variants, we now have a wide variety of tools for inferring structure in network data. Each model will perform better on some networks and worse on others. A better understanding of the strengths and weaknesses of each one—which kinds of structure they can see or they are blind to—will help us select the right algorithm each time we meet a new network.

References

[1] L.A. Adamic and N. Glance. The political blogosphere and the 2004 US election: divided they blog. *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

[2] E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

[3] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 2002.

[4] Stefano Allesina and Mercedes Pascual. Food web models: a plea for groups. *Ecology letters*, 12(7):652–662, 2009.

[5] H. Bauke, C. Moore, J.-B. Rouquier, and D. Sherrington. Topological phase transition in a network model with preferential attachment and node removal. *European Physical Journal B*, 83:519–524, 2011.

[6] A. Clauset, C.R. Shalizi, and M.E.J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51:661–703, 2009.

[7] Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[8] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6), 2011.

[9] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Inference and Phase Transitions in the Detection of Modules in Sparse Networks. *Physical Review Letters*, 107(6), 2011.

[10] S.E. Fienberg and S. Wasserman. Categorical data analysis of single sociometric relations. *sociological Methodology*, pages 156–192, 1981.

[11] W.N. Francis and H. Kucera. Brown Corpus Manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979.

[12] P.W. Holland, K.B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[13] B. Karrer and M. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), 2011.

[14] C. Moore, X. Yan, Y. Zhu, J.-B. Rouquier, and T. Lane. Active learning for node classification in assortative and disassortative networks. In *Proc. 17th KDD*, pages 841–849, 2011.

[15] Cristopher Moore, Gourab Ghoshal, and M. E. J. Newman. Exact solutions for models of evolving networks with addition and deletion of nodes. *Phys. Rev. E*, 74:036121, 2006.

[16] M. Mørup and L.K. Hansen. Learning latent structure in complex networks. *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009.

[17] M. Newman and E.A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007.

[18] M.E. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002.

[19] M.E. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[20] M.E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[21] Jörg Reichardt, Roberto Alamino, and David Saad. The interplay between microscopic and mesoscopic structures in complex networks. *PloS one*, 6(8):e21282, 2011.

[22] T.A. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

[23] S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9(1):1–36, 1987.

[24] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.