arXiv:1404.0431v2 [stat.ML] 3 Jun 2014

Learning Latent Block Structure in Weighted Networks

Christopher Aicher
Department of Applied Mathematics, University of Colorado, Boulder, CO, 80309
christopher.aicher@colorado.edu

Abigail Z. Jacobs
Department of Computer Science, University of Colorado, Boulder, CO, 80309

Aaron Clauset
Department of Computer Science, University of Colorado, Boulder, CO, 80309
BioFrontiers Institute, University of Colorado, Boulder, CO 80303
Santa Fe Institute, Santa Fe, NM 87501

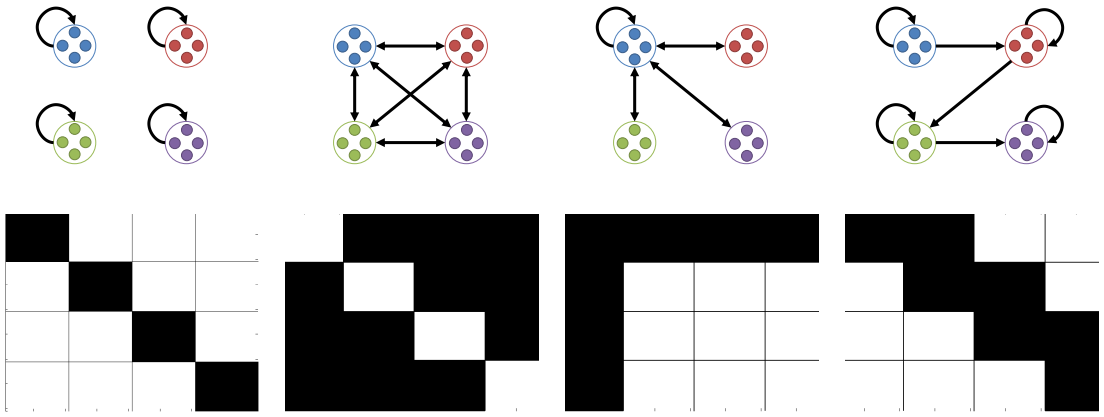
Abstract

Community detection is an important task in network analysis, in which we aim to learn a network partition that groups together vertices with similar community-level connectivity patterns. By finding such groups of vertices with similar structural roles, we extract a compact representation of the network's large-scale structure, which can facilitate its scientific interpretation and the prediction of unknown or future interactions. Popular approaches, including the stochastic block model, assume edges are unweighted, which limits their utility by discarding potentially useful information. We introduce the *weighted stochastic block model* (WSBM), which generalizes the stochastic block model to networks with edge weights drawn from any exponential family distribution. This model learns from both the presence and weight of edges, allowing it to discover structure that would otherwise be hidden when weights are discarded or thresholded. We describe a Bayesian variational algorithm for efficiently approximating this model's posterior distribution over latent block structures. We then evaluate the WSBM's performance on both edge-existence and edge-weight prediction tasks for a set of real-world weighted networks. In all cases, the WSBM performs as well or better than the best alternatives on these tasks. community detection, weighted relational data, block models, exponential family, variational Bayes.

1 Introduction

Networks are an increasingly important form of structured data consisting of interactions between pairs of individuals in large social and biological data sets. Unlike attribute data where each observation is associated with an individual, network data is represented by graphs, where individuals are vertices and interactions are edges. Because vertices are pairwise related, network data violates traditional assumptions of attribute data, such as independence. This intrinsic difference in structure prompts the development of new tools for handling network data.

In social and biological networks, vertices often play distinct structural roles in generating the network's large-scale structure. To identify such latent structural roles, we aim to identify a network partition that groups together vertices with similar group-level connectivity patterns. We call these groups "communities," and their inference produces a compact description of the large-scale



(a) Assortative (b) Disassortative (c) Core-Periphery (d) Ordered

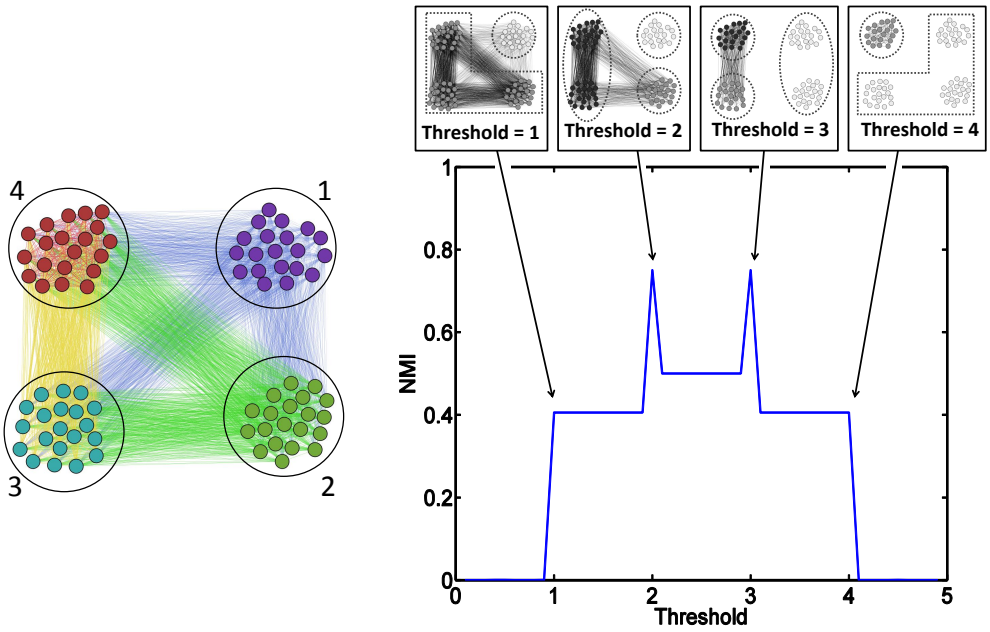
Figure 1: Examples of structure that can be learned using the SBM. The first row shows the abstract connections between four groups (blue, red, green, and purple). The second row shows the 'block' structure found in the adjacency matrix after sorting by group membership; black corresponds to edges and white corresponds to non-edges. (a) Assortative structure: edges mainly exist within groups. (b) Disassortative structure: edges mainly exist between distinct groups. (c) Core-Periphery structure: the 'core' (blue) connects mainly with itself and the 'periphery' (red, green, and purple), while the 'periphery' mainly connects with the 'core'. (d) Ordered structure: blue connects to red, red connects to green, and green connects to purple.

structure of a network. (We note that this definition of a "community" is more general than the assortative-only definition that is commonly used.) This compact large-scale description itself has many potential uses, including dividing a large heterogeneous system into several smaller and more homogeneous parts that may be studied semi-independently, and in predicting unknown or future patterns of interactions. By grouping vertices by these roles, community detection in networks is similar to clustering in vector spaces, and many approaches have been proposed [13].

The stochastic block model (SBM) [17, 34] is a popular generative model for learning community structure in unweighted networks. In its classic form, the SBM is a probabilistic model of pairwise interactions among n vertices. Each vertex i belongs to one of K latent groups or "blocks" denoted by z_i, and each edge A_{ij} exists with a probability $\theta_{z_i z_j}$ that depends only on the group memberships of the connecting vertices. Vertices in the same block are stochastically equivalent, indicating their equivalent roles in generating the network's structure. The SBM is fully specified by a vector z denoting the group membership of each vertex and a $K \times K$ matrix θ of edge bundle probabilities, where $\theta_{k,k'}$ gives the probability that a vertex in group k connects to some vertex of group k'.

The SBM is popular in part because it can generate a wide variety of large-scale patterns of network connectivity depending on the choice of θ (Figs 1(a-d)). For example, if the diagonal elements of θ are greater than its off-diagonal elements, the block structure is assortative, with communities exhibiting greater edge densities within than between them (Fig. 1(a))—a common pattern in social networks [21]. Reversing the pattern in θ generates disassortative structure (Fig. 1(b)), which is often found in language and ecological networks [22]. Other choices of θ can generate hierarchical, multi-partite, or core-periphery patterns [9, 26]. The SBM also has been generalized for count-valued data, degree-correction [18], bipartite structure [19], and categorical values [15].

In addition to this flexibility, the SBM's probabilistic structure provides a principled approach to



(a) Example Network

(b) NMI vs Threshold

Figure 2: (a) An example of a weighted network where thresholding will never succeed. (b) A plot of the normalized mutual information (NMI) between the true community structure and inferred SBM community structure after thresholding at various threshold values (averaged over 100 trials). Examples of community structure found by thresholding are shown above the graph (different colors represent different communities). As the NMI is less than 1 for all threshold values, the SBM after thresholding never infers the true community structure shown in (a).

quantifying uncertainty of group membership, an attractive feature in unsupervised network analysis. This structure has led to theoretical guarantees, including consistency of the SBM estimators [7] and the identifiability and consistency of latent block models [3, 4].

However, each of these models assumes an unweighted network, where edge presence or absence is represented as a binary variable (or perhaps a count-valued variable), while most real-world networks have weights, e.g., interaction frequency, volume, or character. Such information is typically discarded via thresholding before analysis, which can obscure or distort latent structure [33]. To illustrate this loss of information from thresholding, consider a toy network of four equally-sized groups labeled 1–4 (see Fig. 2), where each edge (i, j) is assigned a weight equal to the smaller of the endpoints' group labels, plus a small amount of noise. Edges between groups are thus assigned weights near 1, 2, or 3, while those within a group are assigned weights near 1–4. This model is obviously unrealistic, but serves to illustrate the common consequences of applying a global threshold to edge-weighted networks.

To apply the SBM to this simple network, we must convert it into an unweighted network by discarding edges with weights less than some threshold. To illustrate the results of this action, we consider all possible thresholds, and compute the average normalized mutual information (NMI) between the best community structure found using the SBM and the true structure (Fig. 2). No matter what threshold we choose, edges are divided into at most three groups: those with weight

above, at, or below the threshold. The SBM can thus recover a maximum of three groups, rather than the four planted in this network, and the threshold determines which three groups it finds. No threshold yields the correct inference here, because thresholding discards edge weight information.

Instead of thresholding, we could use more complex methods, such as using multiple thresholds or a binning scheme, to convert a weighted network into an unweighted or count-valued network of some sort. These methods would perform better than applying a single threshold, at the cost of additional complexity in specifying multiple threshold or bin values. Regardless of the method, these approaches will still discard potentially useful edge weight information. To exploit the maximal amount of information in the original data in recovering the true hidden structure, we should prefer to model the edge weights directly.

In this paper, we introduce the *weighted stochastic block model* (WSBM), a generalization of the SBM that can learn from both the presence and weight of edges. The weighted stochastic block model provides a natural solution to this problem by generalizing the SBM to learn from both types of edge information. Specifically, the WSBM models each weighted edge A_{ij} as a draw from a parametric exponential family distribution, whose parameters depend only on the group memberships of the connecting vertices i and j. It includes as special cases most standard distributional forms, e.g., the normal, the exponential, and their generalizations, and enables the direct use of weighted edges in recovering latent group or block structure. This paper generalizes and extends our previous work [1].

We first describe the form of the WSBM, which combines edge existence and weight information. We then derive a variational Bayes algorithm for efficiently learning WSBM parameters from data. Applying this algorithm to a small real-world weighted network, we show that the SBM and WSBM can learn distinct latent structures as a result of observing or ignoring edge weights. Finally, we compare the performance of the WSBM to alternative methods for two edge prediction tasks, using a set of real-world networks. In all cases, the WSBM performs as well as alternatives on edge-existence prediction, and outperforms all alternatives on edge-weight prediction. This model thus enables the discovery of latent group structures in a wider range of networks than was previously possible.

2 Weighted Stochastic Block Model

We begin by reviewing the SBM and exponential families, and then describe a natural generalization of the SBM to weighted networks. In what follows, we consider the general case of directed graphs; undirected graphs are a special case of this model.

In the SBM, the network's adjacency matrix A contains binary values representing edge existences, i.e., $A_{ij} \in \{0, 1\}$, the integer K denotes a fixed number of latent groups, and the vector z contains the group label of each vertex $z_i \in \{1, \ldots, K\}$. The number of latent groups K controls the model's complexity and may be chosen in a variety of ways—we defer a discussion of this matter until section 3.3. Each possible group assignment vector z represents a different partition of the vertices into K groups, and each pair of groups (kk') defines a "bundle" of edges that run between them. The SBM assigns an edge existence parameter to each edge bundle $\theta_{kk'}$, which we represent collectively by the K-by-K matrix θ. The existence probability of an edge A_{ij} is given by the parameter $\theta_{z_i z_j}$ that depends only on the group memberships of vertices i and j.

Assuming that each edge existence A_{ij} is conditionally independent given z and θ, the SBM's likelihood function is

$$\Pr(A \mid z, \theta) = \prod_{ij} \theta_{z_i z_j}^{A_{ij}} \left(1 - \theta_{z_i z_j}\right)^{1 - A_{ij}} \quad , \tag{1}$$

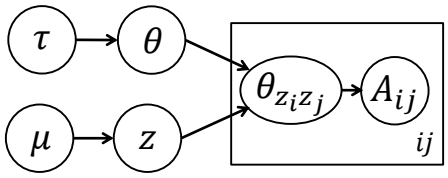


Figure 3: Graphical model for the WSBM. Each weighted edge A_{ij} (plate) is distributed according to the appropriate edge parameter θ_{z_i,z_j} for each observed interaction (i,j). In our variational Bayes inference scheme, the WSBM's latent parameters z, θ are themselves modeled as random variables distributed according to μ, τ, respectively. We highlight that the arrow from z to θ_{z_i,z_j} hides the complex relational structure between each z_i.

which we may rewrite as

$$\Pr(A \mid z, \theta) = \prod_{ij} \exp\left(A_{ij} \cdot \log\left(\frac{\theta_{z_i z_j}}{1 - \theta_{z_i z_j}} \right) + \log\left(1 - \theta_{z_i z_j}\right) \right) \quad .$$

Thus, the likelihood has the form of an exponential family

$$\Pr(A \mid z, \theta) \propto \exp\left(\sum_{ij} T(A_{ij}) \cdot \eta(\theta_{z_i z_j}) \right) \quad , \tag{2}$$

where $T(x) = (x, 1)$ is the vector-valued function of sufficient statistics of the Bernoulli random variable and $\eta(x) = (\log[x/(1-x)], \log[1-x])$ is the vector-valued function of natural parameters. Appendix B provides further details about exponential families.

This choice of functions (T, η) produces binary-valued edge weights. By choosing an appropriate but different pair of functions (T, η), defined on some domain $\mathcal{X}$ and $\times$ respectively, we may specify a stochastic block model whose weights are drawn from an exponential family distribution over $\mathcal{X}$. As in the SBM, this weighted stochastic block model (WSBM) is defined by a vector z and matrix θ, but now each $\theta_{z_i z_j}$ specifies the parameters governing the weight distribution of the $(z_i z_j)$ edge bundle. Figure 3 visualizes the dependencies in the WSBM's likelihood function as a graphical model.

The generative process of creating a weighted network from the WSBM consists of the following steps.

- For each vertex i, assign a group membership z_i.
- For each pair of groups (k, k'), assign an edge bundle parameter $\theta_{kk'} \in \times$
- For each edge (i, j), draw $A_{ij} \in \mathcal{X}$ from the exponential family (T, η) parametrized by $\theta_{z_i z_j}$.

The community structure of the WSBM retains the stochastic equivalence principle of the classic SBM, in which all vertices in a group maintain the same probabilistic connectivity to the rest of the network.

For example, if the edge weights are real-valued $\mathcal{X} = \mathbb{R}$, then we may choose to model the edge weights with the normal distribution, which has sufficient statistics $T = (x, x^2, 1)$ and natural

parameters $\eta = (\mu/\sigma^2, -1/(2\sigma^2), -\mu^2/(2\sigma^2))$. Instead of edge-existence probabilities, each edge-bundle $(z_i z_j)$ is now parameterized by a mean and variance $\theta_{z_i z_j} = (\mu_{z_i z_j}, \sigma^2_{z_i z_j})$. In this case, the likelihood function would be

$$\Pr(A \mid z, \mu, \sigma^2) = \prod_{ij} \mathcal{N}\left(A_{ij} \mid \mu_{z_i z_j}, \sigma^2_{z_i z_j}\right) = \prod_{ij} \exp\left(A_{ij} \cdot \frac{\mu_{z_i z_j}}{\sigma^2_{z_i z_j}} - A^2_{ij} \cdot \frac{1}{2\sigma^2_{z_i z_j}} - 1 \cdot \frac{\mu^2_{z_i z_j}}{\sigma^2_{z_i z_j}}\right) \quad . \quad (3)$$

That is, this particular WSBM uses a normal distribution instead of a Bernoulli distribution to model the values observed in an edge bundle. We emphasize that the choice of the normal distribution is merely illustrative.

This construction produces complete graphs, in which every pair of vertices is connected by an edge with some real-valued weight. For a complete network, this formulation may be entirely sufficient. However, most real-world networks are sparse, with only $O(n)$ pairs having a connection that may have a weight, and a dense model like this one cannot be applied directly. We now describe how sparsity can be naturally incorporated within our model, which also produces more scalable inference algorithms.

2.1 Sparse Weighted Graphs

A key insight for modeling edge-weighted sparse networks lays in clarifying the meaning of zeros in a weighted adjacency matrix. Typically, a value $A_{ij} = 0$ may represent one of three things: (i) the absence of an edge, (ii) an edge that exists but has weight zero, or (iii) missing data, i.e., an unobserved interaction. In both of the former two cases, we do in fact observe the interaction, while in the latter, we do not. For observed interactions, we call the observed non-interaction to be a "non-edge," and we let $A_{ij} = 0$ denote the presence of an edge with weight zero. In many empirical networks, distinct types of interactions may have been confounded, e.g., non-edges, edges with zero weight, and unobserved interactions may all be assigned a value $A_{ij} = 0$. However, for accurate inference, this distinction can be important. For example, a non-edge may indicate an interaction that is impossible to measure, which is distinct from choosing not to measure the interaction (an unobserved interaction) or an interaction with weight zero.

Here, we assume that these three types of interactions are distinguished in our input data. This creates two types of information: information from edge existence (non-edges vs weighted edges) and information from edge weight (the weighted values). To handle these two types of information, the WSBM then models an edge's existence as a Bernoulli or binary random variable, as in the SBM, and models an edge's weight using an exponential family distribution. Terms corresponding to unobserved interactions contribute no information to inference and are dropped from the likelihood function. If the pair (T_e, η_e) denotes the family of edge-existence distributions and the pair (T_w, η_w) denotes the family of edge-weight distributions then we may combine their contributions in the likelihood function via a simple tuning parameter $\alpha \in [0, 1]$ that determines their relative importance in inference

$$\log \Pr(A \mid z, \theta) = \alpha \sum_{ij \in E} T_e(A_{ij}) \cdot \eta_e\left(\theta^{(e)}_{z_i z_j}\right) + (1 - \alpha) \sum_{ij \in W} T_w(A_{ij}) \cdot \eta_w\left(\theta^{(w)}_{z_i z_j}\right) \quad , \quad (4)$$

where E is the set of observed interactions (including non-edges) and W is the set of weighted edges ($W \subset E$). This generalization can be reduced to the compact form of Eq. (2) by combining the vectors αT_e with $(1 - \alpha) T_w$ and η_e with η_w.

By tuning α, we can learn different latent structures. When $\alpha = 1$, the model ignores edge weight information and reduces to the SBM. When $\alpha = 0$, the model treats edge absence as if it

were unobserved, and fits only to the weight information. When $0 < \alpha < 1$, the likelihood combines information from both edge existence and weights. In principle, the best choice of α could also be learned, but we leave this subtle problem for future work. In practice, we often find that $\alpha = 1/2$, giving equal weight to both types of information, works well.

2.2 Degree Correction

The last piece of the WSBM is a generalization to naturally handle heavy-tailed degree distributions, which are ubiquitous in real-world networks and are known to cause the SBM to produce undesirable results, e.g., placing all high-degree vertices in a group together, regardless of their natural community membership [18].

Karrer and Newman introduced an elegant extension of the SBM that circumvents this behavior. In their "degree corrected" SBM (here DCBM), they add vertex degree information into the generative model by adding an "edge-propensity" parameter ϕ_i to each vertex [18]. As a result, the number of edges that exist between a pair of vertices i and j is a Poisson random variable with mean $\phi_i \phi_j \theta_{z_i, z_j}$. Because vertices with high propensity are more likely to connect than vertices with low propensity, the propensity parameters ϕ allow for heterogenous degree distributions within groups. In the DCBM, vertices in the same block are no longer stochastically equivalent, but have similar group-level connectivity patterns conditioned on their propensity parameters ϕ.

The likelihood function for this model is

$$\Pr(A \mid z, \theta, \phi) \propto \prod_{ij} \left(\phi_i \phi_j \theta_{z_i z_j}\right)^{A_{ij}} \exp\left(-\phi_i \phi_j \theta_{z_i z_j}\right) \quad ,$$

where the maximum likelihood estimate of each propensity parameter ϕ_i is simply the vertex degree d_i [18]. By fixing $\phi_i = d_i$, we can rewrite the DCBM in the exponential family form

$$\Pr(A \mid z, \theta, \phi) \propto \prod_{ij} \exp\big(A_{ij} \cdot \log \theta_{z_i z_j} - d_i d_j \cdot \theta_{z_i z_j}\big) \quad , \tag{5}$$

where the sufficient statistics are $T = (A_{ij}, -d_i d_j)$ and the natural parameters are $\eta = (\log \theta_{z_i z_j}, \theta_{z_i z_j})$. Thus, to derive a degree-corrected weighted stochastic block model, we simply replace the SBM contribution in Eq. (4) with that of the DCBM in Eq. (5). We note that this model can easily extended to included in- and out-propensity parameters for directed networks.

This degree-corrected weighted stochastic block model allows for heterogeneous degree distributions within groups by modeling vertex degree or rather the sum of edge existences. This is distinct from what one might call a 'strength'-corrected SBM that produces heterogeneous weight distributions within edge bundles by modeling vertex strength (the sum of a vertex's edge weights). This 'strength'-corrected model is not consider here and is an area for future work.

3 Learning Latent Block Structure

Given some sparse weighted graph A, we recover the underlying communities by learning the parameters z, θ. Any of a large number of standard approaches can be used to optimize the likelihood function for the WSBM. Here, we describe an efficient variational Bayes approach [5, 16], which effectively handles one technical difficulty in fitting the model to real data.

Specifically, learning the parameters z, θ by directly maximizing the likelihood in Eq. (2) can suffer degenerate solutions under continuous valued weights. For instance, consider the WSBM with normally distributed edge weights, where some bundle of edges has all-equal weights. In this case,

the maximum likelihood estimate is a variance parameter equal to zero, which creates a degeneracy in the likelihood calculation. This case is not pathological, as a poor choice of partition z—chosen, perhaps, inadvertently over the course of maximizing the likelihood—can easily create two small groups with only a few edges, each with the same weight, between them. This problem has not previously been identified in the block-modeling literature because the SBM is a model where edge "weights" are discrete Bernoulli random variables, whose parameters are never degenerate.

We solve this problem using Bayesian regularization. In the Bayesian framework, we treat the parameters as random variables and assign an appropriate prior distribution π to our parameters z, θ. If we treat the prior distribution as the probability of the parameters $\pi(z, \theta) = \Pr(z, \theta)$ then we may calculate the posterior distribution as the probability of the parameters conditioned on the data $\pi^*(z, \theta) = \Pr(z, \theta|A)$ through Bayes' law

$$\pi^*(z, \theta) \propto \Pr(A|z, \theta)\pi(z, \theta) \ .$$

After calculating the posterior distribution, we may either return our posterior beliefs π^* about the parameters z, θ or further calculate a point estimate to minimize a posterior expected loss with respect to a given loss function [23, 31]. In both cases, it suffices to calculate the posterior π^*. The maximum likelihood estimate corresponds to only maximizing the likelihood $\Pr(A|z, \theta)$. The inclusion of the prior distribution π prevents the posterior distribution π^* from over-fitting to the degenerate maximum likelihood solution and therefore estimation can proceed smoothly.

However, the posterior distribution is generally difficult to calculate analytically. Instead, we approximate $\pi^*(z, \theta)$ by a factorizable distribution $q(z, \theta) = q_z(z)q_\theta(\theta)$, a common approach in both machine learning and statistical physics. We select our approximation q by minimizing its Kullback-Leibler (KL) divergence to the posterior

$$D_{\mathrm{KL}}(q \,||\, \pi^*) = -\int q \, \log \frac{\pi^*}{q} \ .$$

The Kullback-Leibler divergence is a non-symmetric, non-negative, information-theoretic measure of difference between two distribution. Thus, our approximation q can be thought of as the closest approximation to the posterior π^*, subject to factorization and distribution constraints.

Expanding the constant likelihood $\log \Pr(A)$, we observe that minimizing the KL-divergence is equivalent to maximizing the functional $\mathcal{G}(q)$ defined as follows. Let

$$\begin{aligned}
\log \Pr(A) &= \int_\Theta \sum_{z \in Z} q(z, \theta) \ d\theta \ \ \log \Pr(A) \\
&= \int_\Theta \sum_{z \in Z} q(z, \theta) \log \frac{\Pr(A, z, \theta)}{\Pr(z, \theta|A)} \ d\theta \\
&= \int_\Theta \sum_{z \in Z} q(z, \theta) \log \frac{\Pr(A, z, \theta)}{q(z, \theta)} \ d\theta - \int_\Theta \sum_{z \in Z} q(z, \theta) \log \frac{\Pr(z, \theta|A)}{q(z, \theta)} \ d\theta \\
&= \mathcal{G}(q) + D_{KL} \left(q(z, \theta)||\pi^*(z, \theta)\right) \ ,
\end{aligned}$$

where

$$\mathcal{G}(q) = \int_\Theta \sum_{z \in Z} q(z, \theta) \log \frac{\Pr(A, z, \theta)}{q(z, \theta)} \ d\theta = \mathbb{E}_q(\log \Pr(A \,|\, z, \theta)) + \mathbb{E}_q\left(\log \frac{\pi(z, \theta)}{q(z, \theta)}\right) \ . \tag{6}$$

The first term of Eq. (6) is the expected log-likelihood under the approximation q and the second term is the negative KL-divergence of the approximation q from the prior π. Therefore, we aim to

maximize the expected log-likelihood of the data and weakly constrain the approximation to be close to the prior. The second term serves as a regularizer which prevents over-fitting and eliminates the aforementioned maximum likelihood degeneracies. In practice, the first term dominates the second term given sufficient data and approximates the maximum likelihood estimation.

Because the KL-divergence is non-negative, we can think of $\mathcal{G}(q)$ as a functional lower bound on the log-evidence or marginal log-likelihood, that is,

$$\log \Pr(A) = \mathcal{G}(q) + D_{\mathrm{KL}}\left(q \,||\, \pi^*\right) \geq \mathcal{G}(q) \ . \tag{7}$$

Maximizing $\mathcal{G}(q)$ is equivalent to minimizing the KL divergence $D_{\mathrm{KL}}(q \,||\, \pi^*)$ because the log-evidence $\log \Pr(A)$ is constant. Therefore as we maximize $\mathcal{G}(q)$, our approximation q gets closer to the true posterior π^*. For more details on variational Bayesian inference in graphical models, we refer the interested reader to Ref. [5].

3.1 Conjugate Distributions

To calculate $\mathcal{G}$ in practice, we must assign prior distributions π to our parameters and place constraints on the distributions of our approximation q. For mathematical convenience, we choose π and restrict q to be the product of parameterized conjugate distributions. Because q takes a parameterized form, maximizing the functional $\mathcal{G}(q)$ over all factorized distributions q simplifies to maximizing $\mathcal{G}(q)$ over the parameters of q.

For the edge bundle parameters θ, the standard conjugate prior of the parameter of an exponential family (T, η) is

$$\pi(\theta) = \frac{1}{Z(\tau)} \exp\left(\tau \cdot \eta(\theta)\right) \ , \tag{8}$$

where τ parameterizes the prior and $Z(\tau)$ is a normalizing constant for fixed τ.

For notational convenience, we let r index into the $K \times K$ edge-bundles between groups; hence $\theta = (\theta_1, ..., \theta_r)$. When we update the prior based on the observed weights in a given edge bundle r, the posterior's parameter becomes $\tau^* = \tau + T_r$, where T_r is the sufficient statistic of the observed edges. Thus τ can be viewed as a set of pseudo-observations that push the likelihood function away from the degenerate cases so that every edge bundle, no matter how small or uniform, produces a valid parameter estimate.

For the vertex labels z, the natural conjugate prior is a categorical distribution with parameter $\mu \in \mathbb{R}^{n \times k}$.The parameter $\mu_i(k)$ represents the probability that vertex i belongs to group k in all of its interactions. If the probability in parameter μ_i is spread among multiple groups, then this indicates uncertainty in the membership of vertex i and not mixed membership. We fit μ_i directly, with flat prior $\mu_0(k) = 1/K$.

The form of our prior is thus

$$\pi(z, \theta \,|\, \mu_0, \tau_0) = \prod_i \mu_0(z_i) \times \prod_r \frac{1}{Z(\tau_0)} \exp(\tau_0 \cdot \eta(\theta_r)) \ , \tag{9}$$

where μ_0, τ_0 are the parameters for the priors π_i, π_r, picked to be a "non-informative" reference prior [6] or flat.

Similarly, our approximation q takes the form

$$q(z, \theta \,|\, \mu, \tau) = \prod_i \mu_i(z_i) \times \prod_r \frac{1}{Z(\tau_r)} \exp(\tau_r \cdot \eta(\theta_r)) \ . \tag{10}$$

3.2 An efficient algorithm for optimizing $\mathcal{G}$

Now we consider maximizing $\mathcal{G}$ over q's parameters μ_i, τ_r. To simplify notation, let $\langle T \rangle_r$, $\langle \eta \rangle_r$ be the expected values of the sufficient statistics T_r and natural parameters η_r under the approximation q, that is, we set

$$\langle T \rangle_r = \sum_{ij} \sum_{(z_i, z_j)=r} \mu_i(z_i)\, \mu_j(z_j)\, T(A_{ij}) \tag{11}$$

$$\langle \eta \rangle_r = \frac{\partial}{\partial \tau} \log Z(\tau) \Big|_{\tau=\tau_r} \quad . \tag{12}$$

Substituting the conjugate prior forms of π, q into $\mathcal{G}$ thus yields

$$\mathcal{G} \propto \sum_r \left(\langle T \rangle_r + \tau_0 - \tau_r \right) \cdot \langle \eta \rangle_r + \sum_r \log \frac{Z(\tau_r)}{Z(\tau_0)} + \sum_i \sum_{z_i} \mu_i(z_i) \log \frac{\mu_0(z_i)}{\mu_i(z_i)} \quad . \tag{13}$$

To optimize $\mathcal{G}$, we take derivatives with respect to q's parameters μ, τ and set them to zero. We iteratively solve for the maximum by updating μ and τ independently.

For the edge bundle parameters τ, the derivative of $\mathcal{G}$ is

$$\frac{\partial \mathcal{G}}{\partial \tau_r} = \left(\langle T \rangle_r + \tau_0 - \tau_r \right) \frac{\partial \langle \eta \rangle_r}{\partial \tau_r} \quad , \tag{14}$$

and setting this equal to zero yields a compact update equation

$$\tau_r = \tau_0 + \langle T \rangle_r \tag{15}$$

for each edge bundle r.

For the vertex label parameters μ, we include Lagrange multipliers λ_i to enforce the constraint $\sum_z \mu_i(z) = 1$. Setting the derivative of $\mathcal{G}$ with respect to μ_i equal to λ_i yields

$$\frac{\partial \mathcal{G}}{\partial \mu_i(z)} = \sum_r \left(\frac{\partial \langle T \rangle_r}{\partial \mu_i(z)} \cdot \langle \eta \rangle_r \right) - \log \mu_i(z) = \lambda_i \quad ,$$

where

$$\frac{\partial \langle T \rangle_r}{\partial \mu_i(z)} := \sum_{z' : (z, z') = r} \sum_{j \neq i} T(A_{ij}) \mu_j(z') \quad .$$

Solving for $\mu_i(z)$ yields a compact update equation

$$\mu_i(z) \propto \exp\left(\sum_r \frac{\partial \langle T \rangle_r}{\partial \mu_i(z)} \cdot \langle \eta \rangle_r \right) \quad , \tag{16}$$

where each μ_i is normalized to a probability distribution. To calculate the μ_i values, we iteratively update each μ_i from some initial guess until convergence to within some numerical tolerance.

Algorithm 1 gives pseudocode for the full variational Bayes algorithm, which alternates between updating the edge-bundle parameters and the vertex label parameters using update equations Eqs. (15, 16). Updating θ is relatively fast. First, we calculate $\langle T \rangle_r$ and τ_r for each edge bundle r and then update each $\langle \eta \rangle_r$, which takes $O(nK^2)$ time. Updating μ is the limiting step of the calculation, as we iteratively update μ until convergence while holding θ fixed. To calculate $\partial \langle T \rangle_r / \partial \mu_i(z)$, each

Algorithm 1 Variational Bayes for WSBM

Input: Edge-weighted network A and Model K, α, T, η
Initialize μ
repeat
  for all $r = 1, \ldots, K^2$ **do**
    Set $\langle T \rangle_r := \sum_{ij} \sum_{(z_i, z_j)=r} \mu_i(z_i) \mu_j(z_j) T(A_{ij})$
    Set $\tau_r := \tau_0 + \langle T \rangle_r$
    Set $\langle \eta \rangle_r := \frac{\partial}{\partial \tau} \log Z(\tau) \big|_{\tau=\tau_r}$
  end for
  repeat
    for all $i = 1, \ldots, n$ **do**
      $\frac{\partial \langle T \rangle_r}{\partial \mu_i(z)} := \sum_{(z,z')=r} \sum_{j \neq i} T(A_{ij}) \mu_j(z')$
      $\mu_i(z) \propto \exp\left(\sum_r \frac{\partial \langle T \rangle_r}{\partial \mu_i(z)} \cdot \langle \eta \rangle_r \right)$
    end for
  until μ converge
until μ, τ converge
return μ, τ

vertex must sum over its connected edges for each edge bundle, which takes $O(d_i K^2)$ time. If m is the total number of edges in the network, then updating μ takes $O(mK^2)$ time. In particular, if the total number of edges in the network is sparse $m = O(n)$, then updating μ takes $O(nK^2)$ time.

In practice, we would run the algorithm to convergence from a number of randomly-chosen initial conditions, and then select the best μ, τ.

In addition to the variational Bayes algorithm above, we derive in Appendix C an efficient loopy belief propagation algorithm [12, 35, 36] for the WSBM on sparse graphs. The loopy belief propagation algorithm creates a more flexible approximation to the posterior distribution than the variational Bayes algorithm, but with a slightly higher computational cost. Small modifications for dealing with sparse weighted networks, are described in Appendix D. Finally, Appendix A describes how to obtain our implementation of these methods.

3.3 Selecting K with Bayes factors

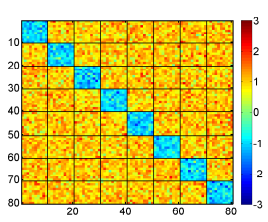
As with most stochastic block models, the number of groups K is a free parameter that must be chosen before the model can be applied to data. For the WSBM, we must also choose the tuning parameter α and the exponential family distributions (T, η).

In principle, any of several model selection techniques could be used, including minimum description length [28], integrated likelihood [11] or Bayes factors [16]. Classic complexity-control techniques like the AIC or BIC are known to misestimate K in certain situations [35]. Here, we describe an approach for choosing K based on Bayes factors that chooses the value K with largest marginal log-likelihood.

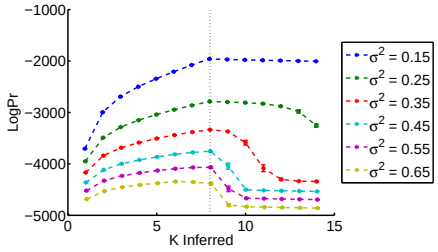
Let $\mathcal{M}_1 = (K_1, \alpha_1, T, \eta)$ and $\mathcal{M}_2 = (K_2, \alpha_2, T, \eta)$ be two competing models, one with K_1 groups and one with K_2 groups. The Bayes factor between these models is

$$\log B(\mathcal{M}_1, \mathcal{M}_2) = \log \frac{\Pr(A \,|\, \mathcal{M}_1)}{\Pr(A \,|\, \mathcal{M}_2)} \approx \mathcal{G}_1 - \mathcal{G}_2 \ , \tag{17}$$

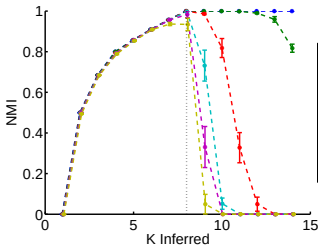
where we approximate the marginal log-likelihood of each model $\Pr(A \,|\, \mathcal{M}_i)$ with our lower bound



(a) Example Network $\sigma^2 = 0.15$



(b) $\mathcal{G}$ vs K Inferred



(c) NMI vs K Inferred

Figure 4: (a) Example network with $K = 8$ groups and variance $\sigma^2 = 0.15$. (b) Approximate marginal log-likelihood $\mathcal{G}$ for each model as a function of K. (c) NMI between the fitted model and the true planted structure as a function of K. Each data series lines in (b,c) corresponds to a different choice of variance σ^2 in edge weights. Results are averaged over 20 trials and the error bars are the standard errors.

$\mathcal{G}_i$ Eq. (7). Although Bayes factors assigns a uniform prior on a set of nested models, this approach has a built-in penalty for complex models through the prior distribution. In our experiments below, we treat K, α, T, η as fixed. This method has produced good results on synthetic data with known planted structure [1].

We now demonstrate the use and efficacy of Bayes factors in selecting the number of groups K in the WSBM. For our demonstration, we choose $K = 8$ groups of 10 vertices each, and consider the method's performance for a variety of edge-weight structures. Specifically, edge weights within each group are drawn from a normal distribution with mean -1 and variance σ^2, while edge weights between groups are drawn from a normal distribution with mean 1 and variance σ^2. By varying the variance parameter σ^2, we vary the difficulty of recovering the true group structure, with a larger variance σ^2 making inference more difficult by causing the edge weight distributions within and between groups to increasingly overlap. Figure 4(a) shows an example network drawn from this model, where we choose $\sigma^2 = 0.15$.

To each choice of σ^2, and for a large number of networks drawn from this model, we fit the WSBM using the normal distribution for the edge weights and vary the number of inferred groups K from 1 to 14. Figure 4(b) shows the approximate marginal log-likelihood $\mathcal{G}$ of each fitted model as K varies, which represents our proportional belief that each choice of K is the correct. Similarly, figure 4(c) shows the NMI between each fitted model and the true planted structure, which represents the performance of each choice of K. Reassuringly, both quantities are maximized at or close to

the true value of K. When the within- and between-group edge-weight distributions are relatively well separated, both the marginal log-likelihood $\mathcal{G}$ and NMI are consistently maximized at $K = 8$, indicating that Bayes factors provide a reasonably reliable method for selecting the correct number of groups and thereby recovering the true planted structure in most cases. As the distributions overlap (greater σ^2 here), it becomes more difficult to distinguish groups, and accuracy degrades to some degree, as would be expected.

4 Experimental Evaluation

In this section, we evaluate the performance of the WSBM on several real-world networks, in two different ways. First, we consider the question of whether adding edge-weight information necessarily reinforces the latent group structure contained in the edge existences. That is, can the WSBM can find structure distinct from what the SBM would find? Second, we evaluate the WSBM's performance on two prediction tasks. The first focuses on predicting missing edges (also called "link prediction"), while the second focuses on predicting missing edge weights. We compare its performance with other block models through cross-validation.

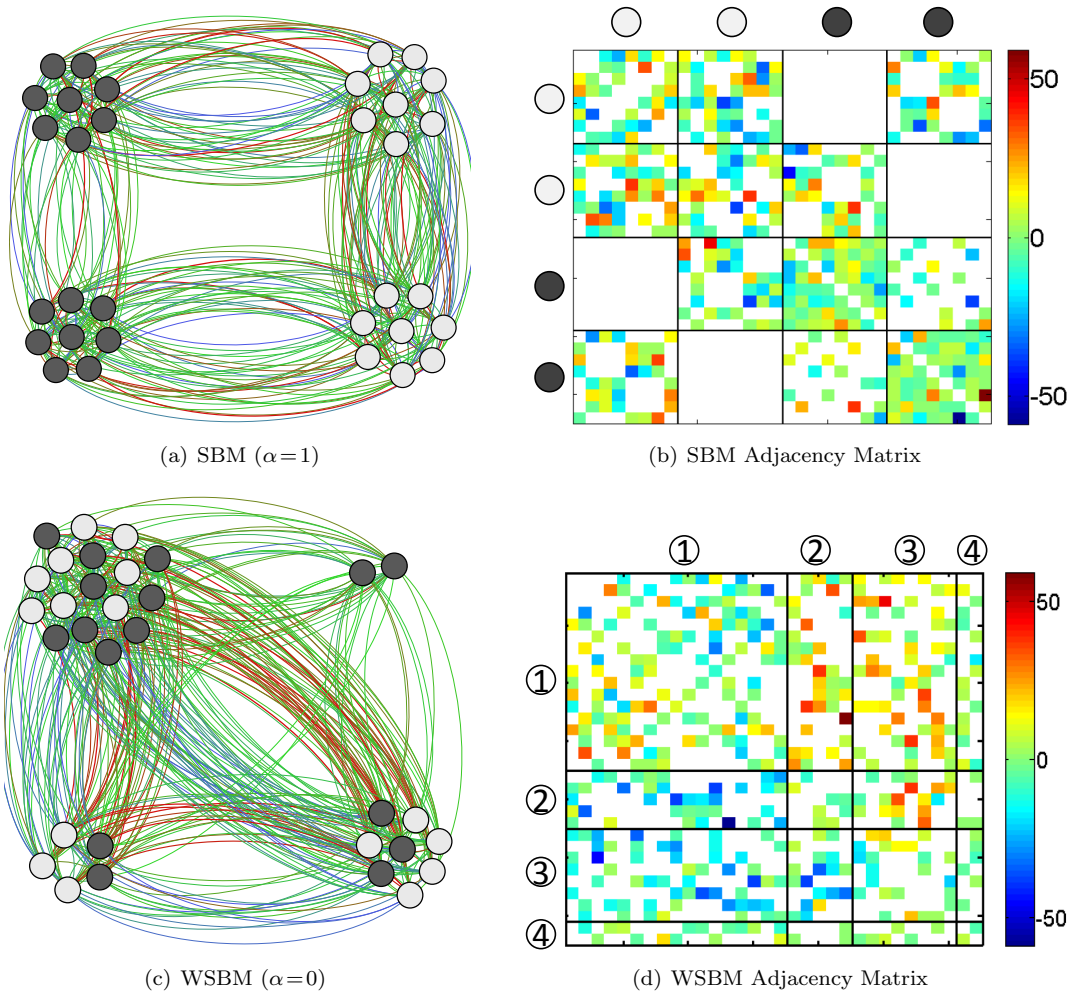
4.1 Edge weight versus edge existence latent group structure

To probe the question of whether edge weights can contain latent group structures that are distinct from those contained in the edge existences, we consider a simple network derived from the competitions among a set of professional sports teams. In this network, called "NFL-2009" hereafter, each vertex represents one of the 32 professional American football teams in the National Football League (NFL). In this network, an edge exists whenever pair of teams played each other in the 2009 season, and each of these edges is assigned a weight equal to the average score difference across games played by that pair [32]. (This definition of edge weight implies the network is skew-symmetric $A_{ij} = -A_{ji}$.) These teams are divided equally among two "conferences" (called AFC and NFC), and within each conference, teams are assigned to one of 4 divisions, each containing 4 teams. Play among teams, i.e., the existence of an edge, is determined by division memberships, and many teams never play each other during the regular season.

To analyze this network, we choose $K = 4$ and fit both the SBM ($\alpha = 1$) and the "pure" WSBM ($\alpha = 0$) using the normal distribution as a model of edge weights. This choice is reasonable for these data as score differences can be positive or negative and score totals are close to a binomial distribution [20]. The $\alpha = 1$ (SBM) case ignores the weights of edges, while the $\alpha = 0$ (pure WSBM) case ignores the presence or absence of edges, focusing only on the observed score differences.

Examining the results of both models, we see that the block structure learned by the SBM ($\alpha = 1$, Figs 5(a-b)) exactly recovers the major divisions within each conference, along with the division between conferences, illustrating that division membership fully explains which teams played each other in this season. That is, the empty off-diagonal blocks (Fig. 5(b)) reflect the fact that two pairs of two divisions never play each other.

In contrast, the block structure learned by the pure WSBM ($\alpha = 0$, Figs 5(c-d)) recovers a global ordering of teams (as in Fig. 1(d)) that reflects each team's general skill, so that teams within each block have roughly equal skill. This pattern mixes teams across conference and division lines, and thus disagrees with the block structure recovered by the SBM. For instance, consider the upper-left group in Fig. 5(c), which generally has positive score differences (wins) in games against teams in either lower group, with a mean lead of 11 points. Similarly, the lower-left group has positive score differences (wins) against teams in the lower-right group. The small upper-right group performs



(a) SBM ($\alpha = 1$)

(b) SBM Adjacency Matrix

(c) WSBM ($\alpha = 0$)

(d) WSBM Adjacency Matrix

Figure 5: NFL-2009 network: black nodes (•) are teams in conference 1 (NFC) and white nodes (○) are teams in conference 2 (AFC). Edges are colored by score differential (red positive, green approximately zero, blue negative). (a) Network showing SBM communities. (b) Adjacency matrix, sorted by SBM communities. (c) Network showing WSBM communities. (d) Adjacency matrix, sorted by WSBM communities. The SBM ($\alpha = 1$) groups correspond to NFL conference structure whereas the WSBM ($\alpha = 0$) corresponds to relative skills levels.

equally well against teams of every other group. Within each group, however, score differences tend toward zero, indicating roughly equal skill.

The fact that the SBM and pure WSBM recover entirely distinct block structures illustrates that adding edge-weight information to the inference step can dramatically alter our conclusions about the latent block structure of a network. That is, adding edge weights does not necessarily reinforce the inferences produced from binary edges alone. The extremal settings of the parameter α in our model allows a practitioner to choose which of these types of latent structure to find, while

if a mixed-type conclusion is preferred, an intermediate value of α may be chosen. In the following section, we demonstrate that such a model, which we call the "balanced" WSBM, that can learn simultaneously from edge existence and weight information.

4.2 Predicting edge existence and weight

To illustrate a more rigorous evaluation of the WSBM, in this section, we consider the problem of predicting missing information when the model is fitted to a partially observed network. In particular, we consider predicting the existence or the weight of some unobserved interaction, a similar task to missing and spurious link prediction [8, 14].

Here, we compare the WSBM to other block models on five real-world networks from various domains. Most of these models are only defined for unweighted networks, and thus some care is required to make them perform under the edge-weight prediction task, which we describe below. We evaluate performance numerically across multiple trials of cross-validation, training each model on 80% of the n^2 possible edges and testing on the remaining 20%.

The weighted graphs we consider are the following.

- *Airport.* Vertices represent the $n = 500$ busiest airports in the United States, and each of the $m = 5960$ directed edges is weighted by the number of passengers traveling from one airport to another [10].
- *Collaboration.* Vertices represent $n = 226$ nations on Earth, and each of the $m = 20616$ edges is weighted by a normalized count of academic papers whose author lists include that pair of nations [25].
- *Congress.* Vertices represent the $n = 163$ committees in the 102nd United States Congress, and each of the $m = 26569$ edges is weighted by the pairwise normalized "interlock" value of shared members [30].
- *Forum.* Vertices represent $n = 1899$ users of a student social network at UC Irvine, and each of the $m = 20291$ directed edges is weighted by the number of messages sent between users [24].
- *College FB.* Vertices represent the $n = 1411$ NCAA college football teams, and each of the $m = 22168$ edges are weighted by the average point difference across games between a pair of teams [32].

For each of the two prediction tasks and for each network, we evaluate the following models. The "pure" WSBM (pWSBM), using only weight information ($\alpha = 0$), a "balanced" WSBM (bWSBM), using both edge and weight information ($\alpha = 0.5$), the "classic" SBM, using only edge information ($\alpha = 1$), a degree-corrected weighted block model DCWBM, where ($\alpha = 0.5$) and the degree-corrected block model (DCBM). For the weighted block models, we select the normal distribution to model the edge weights.

In both prediction tasks, we first choose a uniformly random 20% of the n^2 interactions, which we treat as missing when we fit the model to the network. We then fit each model to the observed edges and infer group membership labels for each vertex in the network. Finally, we use the posterior mean obtained from variational inference as the predictor for edge existence and edge weight for unobserved interactions between those groups. For the models that do not naturally model edge weights (SBM, DCBM), we take their partitions and compute the sample mean weight for each of the induced edge bundles in the weighted network and use this value to predict the weight of any missing edge in that bundle. These estimators correctly correspond to the underlying generative model for edge-prediction in the SBM and DCBM, and are a natural extension for predicting edge-weights for a given block membership. Under this scheme, each model is made to predict the unobserved interactions for a given network, and we score the accuracy of these predictions using the mean-squared error (MSE). Evaluating edge-existence prediction could be achieved using alternative criteria such as

Table 1: Average mean-squared error (MSE) on edge prediction in 25 trials.

Network	pWSBM	bWSBM	SBM	DCWBM	DCBM
Airport	0.0202(1)	**0.0156(1)**	**0.0158(1)**	0.0238(1)	0.0238(1)
Collaboration	0.1446(3)	0.1167(3)	**0.1138(3)**	0.2289(5)	0.2454(5)
Congress	0.1765(4)	**0.1648(4)**	**0.1640(5)**	0.2298(9)	0.2402(9)
Forum	0.00560(1)	**0.00535(1)**	**0.00535(1)**	0.00565(1)	0.00565(1)
College FB	0.0369(2)	**0.0344(1)**	**0.0346(1)**	0.0387(2)	0.0389(2)

Table 2: Average mean-squared error (MSE) on normalized weight prediction in 25 trials.

Network	pWSBM	bWSBM	SBM	DCWBM	DCBM
Airport*	**0.0486(6)**	0.0543(5)	0.0632(8)	0.0746(9)	0.0918(8)
Collaboration*	**0.0407(1)**	0.0462(1)	0.0497(3)	0.0500(2)	0.0849(3)
Congress*	**0.0571(4)**	0.0594(4)	0.0634(6)	0.0653(4)	0.1050(6)
Forum*	**0.0726(3)**	0.0845(3)	0.0851(4)	0.0882(4)	0.0882(4)
College FB	**0.0124(1)**	0.0140(1)	0.0145(1)	0.0149(1)	0.0160(2)

AUC [9], which gives similar results.

Each of these models has a free parameter K that determines the number of parameters that are estimated, which thus controls their overall flexibility. We control this variable model complexity and ensure a fair comparison by fixing all models to have $K=4$ latent groups, and we treat all networks as directed. Finding the true number of latent groups K for each network is separate worthwhile problem not considered here. To compare the results across different data sets, all edge-weights were normalized to fall on the interval $[-1, 1]$. Non-negative weights were normalized after applying a logarithmic transform (cases marked with a star $*$ in Tables 1 and 2).

For each model and each network, we ran 25 independent trials with our 80/20 cross-validation split, as described above, and then compute the average MSE on the particular prediction task. The results for predicting edge existences are summarized in Table 1 and the results for predicting edge weights are summarized in Table 2. Bolded values denote the best MSE across all models, and parentheses indicate the uncertainty (standard error) in the last digit.

Notably, in the edge-existence prediction task, the SBM and the balanced WSBM are the most accurate among all models, often by a large margin. The fact that the SBM performs well is perhaps unsurprising, as it is, by design, only sensitive to edge existences in the first place. However, the balanced WSBM is learning from both existence and weight information, and its strong performance indicates that for these networks, learning from edge weights does not necessarily confuse predictions on edge existence. In the edge-weight prediction task, however, the pure WSBM ($\alpha=0$) is the most accurate, often by a large margin, as we might expect for a model designed to learn only from edge weight information.

In this experimental framework, none of the degree corrected models performs well. This is likely caused by the DCBM's and DCWBM's correction for edge propensity in the group membership. By focusing on finding community structure after accounting for edge propensity, the DCBM and DCWSBM have less accurate predictions in predicting edge existence and edge weight. It is worth pointing out, however, that prediction is not the only measure of utility for community detection techniques, and degree-corrected models often perform better than non-corrected models at recovering meaningful latent group structures in practical situations. We thus expect the degree-

corrected WSBM will be most useful in situations where the goal is the recovery of scientifically meaningful group structures, rather than edge existence or weight prediction.

In general, the SBM performs well on edge prediction but poorly on weight prediction, while the pure WSBM performs poorly on edge prediction but well on weight prediction. This pattern is precisely as we might expect, as the SBM only considers existence information, while the pure WSBM only considers weights.

What is surprising, however, is the good performance on both tasks by the balanced WSBM ($\alpha = 0.5$), which is as good or nearly as good as SBM in edge prediction, but substantially better than the SBM in weight prediction. This demonstrates that the balanced WSBM is a more powerful model than the SBM: it performs as well as the SBM on SBM-like tasks and better on edge weight tasks. In these examples, incorporating edge weight information into the SBM framework does not detract the WSBM performance in edge prediction. In fact, this good general performance is possible because the balanced WSBM learns from both edge existence and edge weight information.

5 Discussion

In the analysis of networks, the inference of latent community structure is a common task that facilitates subsequent analysis, e.g., by dividing a large heterogeneous network into a set of smaller, more homogeneous subgraphs, and can reveal important insights into its basic organizational patterns. When edges are annotated with weights, this extra information is often discarded, e.g., by applying a single universal threshold to all weights. The weighted stochastic block model (WSBM) we described here is a natural generalization of the popular stochastic block model (SBM) to edge-weighted sparse networks. Crucially, the WSBM provides a statistically principled solution to the community detection problem in edge-weighted networks, and removes the need to apply any thresholds before analysis. Thus, this model preserves the maximal amount of information in such networks for characterizing their large-scale structure.

The WSBM's general form, given in Eq. (4), is parametrized by a mixing parameter α, which allows it to learn simultaneously from both the existence (presence or absence) of edges and their associated weights. In our tests with real-world networks, the WSBM yields excellent results on both edge existence and weight prediction tasks. Additionally, the balanced model ($\alpha = 0.5$) performed as well or nearly as well as the best alternative block model, suggesting it may work well as a general model for novel applications where it is not known whether edge existences or edge weights are more informative.

In many applications, the inferred group structure will be of primary interest. For these cases, it is important to note that the groups identified by the WSBM can be distinct from those identified by examining only an unweighted version of the same network. Both forms of latent structure may be interesting and are likely to shed different light on the underlying organization of the network. It remains an open question to determine the types of networks for which weight information contains distinct partition structure from edge existences, although we have shown at least one example of such a network in section 4.1.

The variational algorithm described here provides an efficient method for fitting the WSBM to an empirical network. It scalability is relatively good by modern standards, and thus should be applicable to networks of millions of vertices or more. Alternative algorithms such as those based on Markov chain Monte Carlo for unweighted networks are possible [8, 29]; however, each must contend with several technical problems presented by edge weight distributions, e.g., the degeneracies in the likelihood function produced by edge-bundles whose weights have zero variance.

Finally, there are several natural extensions of the WSBM, including mixed memberships [2], bi-

partite forms [19], dynamic networks [27], different distributions for different edge bundles, and the handling of more complex forms of auxiliary information, e.g., on the vertices or edges. An important and open theoretical question presented by this model is whether utilizing weight information modifies the fundamental detectability of latent group structure, which exhibits a phase transition in the classic SBM [12]. We look forward to these and other extensions.

Funding

This work was supported by the U.S. Air Force Office of Scientific Research and the Defense Advanced Research Projects Agency [grant number FA9550-12-1-0432].

Acknowledgments

We thank Dan Larremore, Leto Peel, and Nora Connor for helpful conversations and suggestions.

Certain data included herein are derived from the Science Citation Index Expanded, Social Science Citation Index and Arts & Humanities Citation Index, prepared by Thomson Reuters, Philadelphia, Pennsylvania, USA, Copyright Thomson Reuters, 2011

A Code Availability

A working implementation of the WSBM inference code, written by the authors, may be found at `http://tuvalu.santafe.edu/%7Eaaronc/wsbm/`.

This code implements the efficient algorithms discussed in Appendix D.

B Exponential Families

Let $\mathcal{X}$ be a fixed domain and Θ be set of parameters. An exponential family is a collection of parametric distributions $\mathcal{F}$ that can be written in the form

$$\mathcal{F} = \{f(x \,|\, \theta) = h(x) \exp\left(T(x) \cdot \eta(\theta)\right) \text{ for } x \in \mathcal{X} \,|\, \theta \in \Theta\} \ ,$$

where h, T, η are fixed functions. The map T is the sufficient statistic function and the map $\eta(\theta)$ are the natural parameters. Note that T and η can be vectors. The function $h(x)$ distinguishes different probability distributions, but appears as an additive constant in the log-likelihood function, which can thus be ignored. Thus, only the pair (T, η) directly impacts the likelihood function.

Examples of exponential families include the normal, exponential, gamma, log-normal, Pareto, binomial, multinomial, Poisson, and beta distributions. Examples of distributions that are not exponential families are the Uniform distribution and certain mixture distributions.

A common representation of an exponential family sometimes includes the log-partition function $A(\theta)$ written as

$$f(x \,|\, \theta) = h(x) \exp\left(\tilde{T}(x) \cdot \tilde{\eta}(\theta) - A(\theta)\right) \ .$$

To keep notation compact we absorb $-1 \cdot A$ into $T \cdot \eta$.

A convenient property of exponential families is that they have easily written conjugate priors. For our exponential family the standard class of conjugate priors π are

$$\pi(\theta) = \frac{1}{Z(\tau)} \exp\left(\tau \cdot \eta(\theta)\right) \ ,$$

where τ are the (hyper-)parameters of the prior and can be thought of as pseudo-observations of T. The function Z is the normalizing constant, defined as

$$Z(\tau) = \int_{\Theta} \exp\left(\tau \cdot \eta(\theta)\right) \, d\theta \ .$$

Finally, it can be shown that the expected value of $\eta(\theta)$ under $\pi(\cdot \,|\, \tau)$ is

$$\langle \eta(\theta) \rangle = \frac{\partial \log Z(\tau)}{\partial \tau} \ .$$

Further details on exponential families can be found in Refs. [23, 31] and for appropriate prior distributions in Ref. [6].

C Belief Propagation Derivation

The main difference between a loopy belief propagation (hereafter simply BP) algorithm and the variational Bayes algorithm described in the main text lays in how we update the group membership

parameters μ [36]. The BP approach gives a more accurate approximation of the true posterior of z and has been shown to produce good results in the classic SBM case [12].

In variational Bayes, we used a mean-field approximation to the posterior distribution π^*:

$$\pi^*(z) \approx q(z) = \prod_i \mu_i(z_i) \ ,$$

where each vertex label is assumed to be independently distributed according to q_i (a categorical or multinomial random variable).

In BP, we use pairwise approximations to the posterior distribution. Ideally, this approximation would have the form

$$\pi^*(z) \approx q(z) \propto \prod_{ij} \mu_{ij}(z_i, z_j) \ ,$$

where $\mu_{ij}(\cdot, \cdot)$ are joint probabilities. However, this typically is not achievable because normalizing the product of distributions over all edge pairs is non-trivial (each vertex of degree k_i appears k_i times). Luckily, in the case of trees, it is possible to normalize q to a probability distribution by accounting for this repetition, that is,

$$q(z) = \frac{\prod_{ij \in E} \mu_{ij}(z_i, z_j)}{\prod_i \mu_i(z_i)^{k_i - 1}} \ ,$$

where μ_i is the marginal of μ_{ij}, k_i is the degree of vertex i, and E is the set of observed edges. But, the factor graph of the WSBM is not a tree, so this form is not necessarily exact.

Here, we take a loopy BP approach and assume the structure of pairwise terms $ij \in E$ is in fact locally tree-like, and then apply the BP update equations. The assumption for locally tree-like structure makes this algorithm a poor choice on dense networks (when we observe $O(n^2)$ interactions), but is both acceptable and effective for sparse networks.

Under this formulation, our goal is to maximize the variational approximation to the likelihood of the data $\mathcal{G}$, so that the KL divergence between q and π^* is minimized. Recall from Eq. (6) the objective function $\mathcal{G}$ consists of two parts

$$\mathcal{G} = \mathbb{E}_q \log \Pr(A \,|\, z, \theta) + \mathbb{E}_q \log (\pi / q) \ ,$$

a likelihood term and a prior regularizer term.

The likelihood term is

$$\mathbb{E}_q \log \Pr(A \,|\, z, \theta) \propto \sum_r \left(\sum_{ij} T(A_{ij}) \mathbb{E}_q(z_i z_j) + \tau_r^{(0)} \right) \cdot \langle \eta \rangle_r \approx \sum_r \left(\langle T \rangle_r + \tau_r^{(0)} \right) \cdot \langle \eta \rangle_r \ ,$$

where

$$\langle T \rangle_r = \sum_{ij} \sum_{(z,z')=r} \mu_{ij}(z, z') \, T(A_{ij})$$
$$\langle \eta \rangle_r = \left. \frac{\partial \log Z(\tau)}{\partial \tau} \right|_{\tau = \tau_r} \ ,$$

and where we approximate $\mathbb{E}_q(z_i z_j) \approx q_{ij}(z_i, z_j)$ and $\tau_r^{(0)}, \mu^{(0)}$ are parameters for the prior.

The regularizer term consists of two parts

$$\mathbb{E}_q \log (\pi / q) = \mathbb{E}_q (\log \pi) - \mathbb{E}_q (\log q) \ .$$

The second term is requires us to sum over $q(z)$ which is combinatorically difficult to calculate, so we use the Bethe approximation

$$-\mathbb{E}_q\left(\log q\right) \approx -\sum_{ij\in E}\sum_{z,z'}\mu_{ij}(z,z')\log\mu_{ij}(z,z')+\sum_{i,z}(k_i-1)\mu_i(z)\log\mu_i(z)+\sum_r -\tau_r\cdot\langle\eta\rangle_r+\log Z(\tau_r)\ .$$

Combining these parts, the objective function may be written as

$$\begin{aligned}\mathcal{G} =& \sum_r\left(\langle T\rangle_r+\tau_r^{(0)}-\tau_r\right)\cdot\langle\eta\rangle_r+\sum_r\log\frac{Z(\tau_r)}{Z(\tau_r^{(0)})}\\ &+\sum_{i,z}(k_i-1)\mu_i(z)\log\frac{\mu_i(z)}{\mu_i^{(0)}(z)}-\sum_{ij\in E}\sum_{z,z'}\mu_{ij}(z,z')\log\frac{\mu_{ij}(z,z')}{\mu_i^{(0)}(z)\mu_j^{(0)}(z')}\ .\end{aligned}$$

To enforce the marginalization and normalization restrictions on $q(z)$, we introduce Lagrange multipliers, yielding

$$\mathcal{G}'=\mathcal{G}+\sum_i\lambda_i\left(\sum_i\mu_i-1\right)+\sum_{ij\in E}\left(\sum_z\lambda_{ij,z}\left(\mu_i(z)-\sum_{z'}\mu_{ij}(z,z')\right)+\sum_{z'}\lambda'_{ij,z'}\left(\mu_j(z')-\sum_z\mu_{ij}(z,z')\right)\right)\ .$$

Note that λ_i enforces normalization of μ_i, $\lambda_{ij,z}$ enforces marginalization over i, and $\lambda'_{ij,z'}$ enforces marginalization over j. We maximize $\mathcal{G}'$ by setting its derivatives with respect to the parameters of q equal to 0

For the edge parameters θ, we differentiate with respect to τ_r

$$\begin{aligned}\frac{\partial\mathcal{G}'}{\partial\tau_r} &= \left(\langle T\rangle_r+\tau_r^{(0)}-\tau_r\right)\frac{\partial\langle\eta\rangle_r}{\partial\tau_r}-\langle\eta\rangle_r+\left.\frac{\partial\log Z(\tau)}{\partial\tau}\right|_{\tau=\tau_r}\\ &\propto \langle T\rangle_r+\tau_r^{(0)}-\tau_r\ .\end{aligned}$$

This is the same expression as for the variational Bayes solution, since we only modified $q(z)$. The update equations for τ remain $\tau_r=\tau_r^{(0)}+\langle T\rangle_r$.

For the vertex labels z, we will differentiate with respect to $\mu_i(z)$ and $\mu_{ij}(z,z')$ and solve this system of equations using a message passing method, which is standard in BP. The derivatives are

$$\frac{\partial\mathcal{G}'}{\partial\mu_i(z)}=(k_i-1)\left(\log\mu_i(z)-\log\mu_i^{(0)}(z)+1\right)+\lambda_i+\sum_{j:ij\in E}\lambda_{ij,z}=0\ ,$$

and

$$\frac{\partial\mathcal{G}'}{\partial\mu_{ij}(z,z')}=T(A_{ij})\cdot\langle\eta\rangle_{z,z'}-\log\mu_{ij}(z,z')+\log\mu_i^{(0)}(z)+\log\mu_j^{(0)}(z')-1-\lambda_{ij,z}-\lambda'_{ij,z'}=0\ .$$

Solving for $\mu_i(z)$ and $\mu_{ij}(z,z')$ we obtain

$$\begin{aligned}\mu_i(z) &\propto \mu_i^{(0)}(z)\prod_{j:ij\in E}e^{-\lambda_{ij,z}/(k_i-1)}\\ \mu_{ij}(z,z') &\propto \mu_i^{(0)}(z)\mu_j^{(0)}(z')\exp\left(T(A_{ij})\cdot\langle\eta\rangle_{z,z'}\right)e^{-\lambda_{ij,z}}e^{-\lambda'_{ij,z'}}\ .\end{aligned}$$

For notational convenience, let

$$M_{ij}(z,z')=\exp\left(T(A_{ij})\cdot\langle\eta\rangle_{z,z'}\right)\ .$$

Since $\sum_{z'} \mu_{ij}(z,z') = \mu_i(z)$, we have

$$\mu_i(z) \propto \mu_i^{(0)}(z) \sum_{z'} \mu_j^{(0)}(z') M_{ij}(z,z') e^{-\lambda_{ij,z}} e^{-\lambda'_{ij,z'}} \ .$$

Setting our two equations for $\mu_i(z)$ are equal, we obtain

$$\mu_i^{(0)}(z) \prod_{j':ij' \in E} e^{-\lambda_{ij',z}/(k_i-1)} \propto \mu_i^{(0)}(z) \sum_{z'} \mu_j^{(0)}(z') M_{ij}(z,z') e^{-\lambda_{ij,z}} e^{-\lambda'_{ij,z'}}$$
$$\prod_{j':ij' \in E} e^{-\lambda_{ij',z}/(k_i-1)} \propto \sum_{z'} \mu_j^{(0)}(z') M_{ij}(z,z') e^{-\lambda_{ij,z}} e^{-\lambda'_{ij,z'}} \ . \qquad (*)$$

Let $\psi_{i\to j}(z_j)$ denote the message from vertex i to vertex j and set

$$e^{-\lambda_{ij,z}} = \prod_{k:ik \in E, k \neq j} \psi_{k\to i}(z)$$

$$e^{-\lambda'_{ij,z'}} = \prod_{k:j,k \in E, k \neq i} \psi_{k\to j}(z') \ .$$

Plugging in our definition of ψ, we obtain

$$\prod_{j:ij \in E} e^{-\lambda_{ij,z}/(k_i-1)} = \prod_{j:ij \in E} \prod_{k:ik \in E, k \neq j} \psi_{k\to i}(z)^{1/(k_i-1)} = \prod_{ij \in E} \psi_{j\to i}(z) \ .$$

And, using Eq. (*), we obtain the following recusive definition for ψ

$$\prod_{ij \in E} \psi_{j\to i}(z) \propto \sum_{z'} \mu_j^{(0)}(z') \left(\sum_{z'} M_{ij}(z,z') \right) \prod_{k:ik \in E, k \neq j} \psi_{k\to i}(z) \prod_{k:jk \in E, k \neq i} \psi_{k\to j}(z')$$
$$\psi_{j\to i}(z) \propto \sum_{z'} \mu_j^{(0)}(z') M_{ij}(z,z') \prod_{k:k,j \in E, k \neq i} \psi_{k\to j}(z') \ .$$

Finally, our update equations for μ become

$$\mu_i(z) \propto \mu_i^{(0)}(z) \prod_{ij \in E} \psi_{j\to i}(z)$$
$$\mu_{ij}(z,z') \propto \mu_i^{(0)}(z) \mu_j^{(0)}(z') M_{ij}(z,z') \prod_{k:ik \in E, k \neq j} \psi_{k\to i}(z) \prod_{l:l,j \in E, l \neq i} \psi_{l\to j}(z') \ .$$

If $m = |E|$ is the number of observed edges/interactions, then the BP algorithm requires $O(m)$ messages to be passed and therefore each iteration has an $O((m+n)K^2)$ running time (updating the messages ψ and then the group membership parameters μ).

It will be convenient to use the following equivalent messages φ used by [12, 35, 37] in our BP algorithm

$$\varphi_{i\to j}(z') = \mu_j^{(0)}(z') \prod_{k:k,j \in E, k \neq i} \psi_{k,j}(z') \ .$$

Note that from our old message ψ update equations, we obtain

$$\psi_{i\to j}(z') = \sum_z M_{ij}(z,z') \varphi_{j\to i}(z) \ .$$

Putting these two equations together, our new update equations using φ for our messages become

$$\varphi_{i\to j}(z') = \mu_j^{(0)}(z') \prod_{k:kj\in E, k\neq i} \sum_z M_{k,j}(z,z')\varphi_{j\to k}(z)$$
$$\mu_i(z) \propto \mu_i^{(0)}(z) \prod_{ij\in E} \sum_{z'} M_{ij}(z,z')\varphi_{i\to j}(z')$$
$$\mu_{ij}(z,z') \propto M_{ij}(z,z')\varphi_{j\to i}(z)\varphi_{i\to j}(z') \ .$$

Algorithm 2 gives pseudocode for the full loopy BP algorithm.

Algorithm 2 Loopy BP for sparse networks

Input: Data E, Model K, α, T, η
Initialize μ
repeat
  for all $r = 1, \ldots, K^2$ **do**
    Set $\langle T \rangle_r := \sum_{ij} \sum_{(z_i,z_j)=r} \mu_i(z_i)\mu_j(z_j)T(A_{ij})$
    Set $\tau_r := \tau_0 + \langle T \rangle_r$
    Set $\langle \eta \rangle_r := \frac{\partial}{\partial \tau} \log Z(\tau)\big|_{\tau=\tau_r}$
  end for
  Calculate M_{ij} for all (ij) in E
  Set $M_{ij}(k,k') = \exp(T(A_{ij}) \cdot \langle \eta \rangle_{k,k'} + T(A_{ji}) \cdot \langle \eta \rangle_{k',k})$ for all k, k'
  repeat
    for all (ij) in E **do**
      Set $\varphi_{j\to i}(z_i) \propto \mu_0(z_i) \prod_{k\neq i, kj\in E} \sum_{z_k} \varphi_{i\to k}(z_k)M_{ik}(z_i,z_k)$
    end for
  until φ converge
  for all $i = 1, \ldots, n$ **do**
    Set $\mu_i(z_i) \propto \mu_0(z_i) \prod_{ij\in E} \sum_{z_j} \varphi_{i\to j}(z_j)M_{ij}(z_i,z_j)$
  end for
until μ, τ converge
return μ, τ

D Modifications for Sparse Weighted Graphs

We now consider modifications to our variational Bayes algorithm (Algorithm 1) and our BP algorithm (Algorithm 2) for the case of sparse weighted graphs discussed in section 2.1.

Recall that for a network of n nodes we can partition the n^2 interaction into 3 disjoint edge lists W, N, M, where W is a list of *weighted edges*, N is a list of *non-edges*, and M is a list of *missing edges* or *unobserved* edges. We define the union $E = W \cup N$ as the list of *observed* edges. Let $m_W = |W|$ be the number of weighted edges, $m_E = |E|$ be the number of observed edges, and $m_M = |M|$ be the number of missing edges. Note that $m_E + m_M = |E| + |M| = |A| = n^2$.

Both algorithms we presented require $O(|E|K^2)$ time when updating μ. If the number of *observed* edges is sparse ($|E| = O(n)$), then no changes are required. However it may be the case that the number of weighted edges is sparse ($|W| = O(n)$), while the number of non-edges is dense

($|N| = O(n^2)$). In this case, if we assume the number of missing edges is also sparse ($|M| = O(n)$), then we can modify Algorithms 1 and 2, so that running time is once again $O(nK^2)$. The key idea is to exploit the structure of our edge-existence distribution.

First we introduce some notation, then we consider the edge bundle τ updates, and finally we introduce modifications to the group membership μ updates.

Notation. There are two types of degrees: the degree with respect to weighted edges and degree with respect to observed edges. Let $d_W^-(i)$ be the in-degree of vertex i with respect to weighted edges. Let $d_W^+(i)$ be the out-degree of vertex i with respect to weighted edges. Let $d_E^-(i)$ be the in-degree of vertex i with respect to observed edges. Let $d_E^+(i)$ be the out-degree of vertex i with respect to observed edges.

Let our exponential family edge-weight distribution f_w under parameter θ_w take form

$$f_w(x \,|\, \theta_w) = h_w(x) \exp\left(T_w(x) \cdot \eta_w(\theta_w)\right) \ ,$$

where h_w, T_w, η_w are fixed functions.

Let our exponential family edge-existence distribution f_e under parameter θ_e take the form

$$f_e(x \,|\, \theta_e) = h_e(x) \exp\left(T_e(x) \cdot \eta_e(\theta_e)\right) \ ,$$

where h_e, T_e, η_e are fixed functions.

Let $R : K \times K \to r$ be the mapping between the groups and edge-bundles.

D.1 Update for τ (edge distribution)

The edge bundle updates consist of two steps: (i) calculating the expected sufficient statistic $\langle T \rangle$ for each edge bundle, and (ii) updating τ for each edge bundle.

Weighted τ_w. For the weighted distribution, the expected sufficient statistic $\langle T_w \rangle_r$ for all edge bundles r can be calculated using Eq. (18) for all pairs of groups (z, z'), as

$$\langle T_w \rangle_{R(z,z')} \mathrel{+}= \sum_{ij \in W} T_w(A_{ij}) \mu_i(z) \mu_j(z') \ . \tag{18}$$

Since the running time for each pair is dominated by the summation over the set W, each iteration over Eq. (18) takes $O(n + m_W)$ in $O(K^2(n + m_W))$ time.

Edge existence τ_e. To update T_e we note that the sufficient statistic value for a non-edge is typically zero except for the last dimension that takes the value 1 for observed edges. Knowing that this value is 1 for all edges lets us calculate T_e without needing to sum over $W \cup N$.

Therefore we update T_e using Eq. (18) for all but the last dimensions of T_e. For the last dimension we update T_e with

$$\langle T_e \rangle_{R(z,z')} \mathrel{+}= \sum_{ij} \mu_i(z) \mu_j(z') - \sum_{ij \in M} \mu_i(z) \mu_j(z') = \left(\sum_i \mu_i(z) \right) \left(\sum_j \mu_j(z') \right) - \sum_{ij \in M} \mu_i(z) \mu_j(z') \ , \tag{19}$$

which takes $O(K^2(n + m_M))$ time.

Degree-corrected edge existence τ_e. For the degree corrected block model, recall that edge existence distribution is modified slightly by replacing the $T_e(A_{ij}) = 1$ in the last dimension of T_e with the product of i, j's in and out degrees, $T_e(A_{ij}) = d_W^+(i)d_W^-(j)$. This changes equation (19) by replacing $\mu_i(z)$ with $d_W^+(i)\mu_i(z)$ and $\mu_j(z')$ with $d_W^-(j)\mu_j(z')$. This gives us

$$\langle T_e \rangle_{R(z,z')} \mathrel{+}= \left(\sum_i d_W^+(i)\mu_i(z)\right)\left(\sum_j d_W^-(j)\mu_j(z')\right) - \sum_{ij\in M} d_W^+(i)\mu_i(z)d_W^-(j)\mu_j(z') \ . \tag{20}$$

The running time remains the same as in the edge existence case.

D.2 Update for μ (vertex labels)

Variational Bayes Algorithm. The update for the vertex labels under the variational Bayes algorithm, is to (i) calculate $\frac{\partial \langle T \rangle_r}{\partial \mu_i(z)}$ and (ii) update μ_i using

$$\mu_i(z) \propto \exp\left(\sum_r \frac{\partial \langle T \rangle_r}{\partial \mu_i(z)} \cdot \langle \eta \rangle_r\right) \ .$$

The rate limiting step is in calculating $\frac{\partial \langle T \rangle_r}{\partial \mu_i(z)}$.

For the weighted sufficient statistics T_w, we calculate for all pairs (z, z') and for each vertex i

$$\frac{\partial \langle T_w \rangle_{R(z,z')}}{\partial \mu_i(z)} \mathrel{+}= \sum_{j\in\partial i_W^+} T_w(A_{ij})\mu_j(z') \quad , \quad \frac{\partial \langle T_w \rangle_{R(z',z)}}{\partial \mu_i(z)} \mathrel{+}= \sum_{j\in\partial i_W^-} \mu_j(z')T_w(A_{ji}) \ , \tag{21}$$

where ∂i_W^+ is the neighborhood formed by the outgoing weighted edges of vetex i. Since the sum in Eq. (21) is over $d_W^+(i)$ terms, the running time is $O(K^2 \sum_i d_W^+(i)) = O(K^2(n + m_W))$.

Similar to how we updated τ_e, in the edge-existence case we update $\frac{\partial \langle T \rangle_r}{\partial \mu_i(z)}$ by calculating the entire sum and subtracting away the missing edges. Again, we exploit the fact that the last dimension of T_e is 1 for observed edges, and

$$\frac{\partial \langle T_e \rangle_{R(z,z')}}{\partial \mu_i(z)} \mathrel{+}= \left(\sum_j \mu_j(z')\right) - \sum_{j\in\partial i_M^+} T_e(A_{ij})\mu_j(z'), \ . \tag{22}$$

Calculating Eq. (22) for all vertices has a total $O(K^2(n + m_M))$ running time if we pre-calculate $\sum_j \mu_j(z')$.

For the degree corrected block model we replace μ_j with $d_W^-(j)\mu_j(z')$ and use Eq. (22).

Loopy BP Algorithm. The update for the vertex labels under the BP algorithm requires us to (i) calculate the marginal evidence from each edge $M_{ij}(z, z')$, (ii) update messages $\varphi_{j\to i}(z_i)$ between weighted edges, (iii) approximate messages $\varphi_{\to i}(z_i) = \mu_i(z_i)$ between non-edges, and (iv) caclulate the vertex label probabilities μ_i.

We calculate the marginal evidence M

$$M_{ij}(z, z') = \exp\left(T(A_{ij}) \cdot \langle \eta \rangle_{R(z,z')} + T(A_{ji}) \cdot \langle \eta \rangle_{R(z',z)}\right) \ , \tag{23}$$

for each weighted edge $ij \in W$ for all z, z'. This takes $O(K^2 m_W)$ time. Note that $M_{ij} = M_{ji}$. For the non-edges, we again exploit the fact that the last dimension of T_e is 1 for observed edges and only need to calculate $M_{ij} = M_N$ once using Eq. (23) for all non-edges $ij \in N$.

The messages between weighted edges are

$$\varphi_{i\to j}(z') \propto \mu_0(z') \prod_{k\neq j, k\in\partial i_W} \sum_{z_k} \varphi_{j\to k}(z_k) M_{jk}(z', z_k) \ . \tag{24}$$

Each step requires $O(|\partial i_W| K^2)$ calculations. In the case of a sparse graph, $\partial i_W = O(1)$ and since we repeat this step for each pair i, j in W, the overall running time is $O(K^2 m_W)$.

Since there are $O(n^2)$ non-edges, the messages between non-edges must be approximated for our algorithm to be efficient. The idea behind this approximation is to exploit the sparsity of the weighted edges.

To be concrete, suppose we select the Bernoulli distribution for our edge-existence distribution $f_e(x \,|\, p)$. Then our marginal evidence takes the form

$$\tilde{M}_{ij}(z, z') = \begin{cases} \exp\left(\langle \log p \rangle_{z,z'}\right) \cdot M_{ij}(z, z') & \text{if } ij \in E \\ \exp\left(\langle \log(1-p) \rangle_{z,z'}\right) & \text{otherwise} \ , \end{cases} \tag{25}$$

where p is the edge-existence parameter θ_e. If the graph is sparse, then $\langle \log(1-p) \rangle_r = O(1/n)$. Thus for $i, j \in E$, we have $\tilde{M}_{ij} \approx 1$. And therefore messages between non-edges can be approximated as

$$\varphi_{i\to j}(z') = \mu_j^{(0)}(z') \prod_{k\neq i} \sum_z \tilde{M}_{k,j}(z, z') \varphi_{j\to k}(z) \approx \mu_j^{(0)}(z') \prod_k \sum_z \tilde{M}_{k,j}(z, z') \varphi_{j\to k}(z) = \mu_j(z') \ . \tag{26}$$

Thus we can approximate all messages between non-edges $\varphi_{i\to j}(z')$ with their marginal distribution $\mu_j(z')$ taking $O(nK)$ space and time.

The Poisson and degree-corrected case are more complicated and should follow along the lines of [35]. This extension is left for future work.

Given the messages and marginal evidence, we calculate the vertex label probabilities with

$$\begin{aligned} \mu_i(z) &\propto \mu_i^{(0)}(z) \prod_{ij\in W} \sum_{z'} M_{ij}(z, z') \varphi_{i\to j}(z') \cdot \prod_{ij\in N} \sum_{z'} M_N(z, z') \mu_j(z') \\ &= \prod_{j\in\partial i_W} \frac{\sum_{z'} M_{ij}(z, z') \varphi_{i\to j}(z')}{\sum_{z'} M_N(z, z') \mu_j(z')} \cdot \left(\sum_{z'} M_N(z, z') \sum_j \mu_j(z') \right)^{\partial i_E} \ , \end{aligned} \tag{27}$$

where ∂i_E is the total (in- and out-) degree of observed edges and M_N is the marginal evidence of a non-edge. Each of these updates takes $O(|\partial i_W| K^2)$ calculations. Let ∂i_W be the in and out neighborhood of i. In the case of a sparse graph, ∂i_W is $O(1)$ and since we repeat this step for each pair i, j in W, the overall running time is $O(K^2 m_W)$.

In conclusion, all three steps take $O(nK^2)$ time when the number of weighted edges and missing edges is sparse ($|W| = O(n)$ and $|M| = O(n)$). Although both the variational Bayes algorithm and the loopy BP algorithm have the same asymptotic running time, the constant in front of $O(nK^2)$ for the loopy BP algorithm depends on the average weighted degree of the network.

References

1. C. Aicher, A. Z. Jacobs, and A. Clauset. Adapting the stochastic block model to edge-weighted networks. *ICML Workshop on Structured Learning (SLG 2013)*, May 2013.
2. E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.
3. E. S. Allman, C. Matias, and J. A. Rhodes. Parameter identifiability in a class of random graph mixture models. *J. Statist. Plann. Inference*, 141(5):1719–1736, May 2011.
4. C. Ambroise and C. Matias. New consistent and asymptotically normal parameter estimates for random-graph mixture models. *J. R. Stat. Soc. Ser. B*, 74(1):3–35, Jan. 2012.
5. H. Attias. A variational bayesian framework for graphical models. In *Adv. in Neural Info. Proc. Sys. 12*, pages 209–215. MIT Press, 2000.
6. J. O. Berger and J. M. Bernardo. On the development of reference priors. *Bayesian statistics*, 4(4):35–60, 1992.
7. A. Celisse, J.-J. Daudin, and L. Pierre. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electron. J. Stat.*, 6:1847–1899, 2012.
8. A. Clauset, C. Moore, and M. E. Newman. Structural inference of hierarchies in networks. In *Lecture Notes in Computer Science*, volume 4503, pages 1–13. Springer, 2007.
9. A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.
10. V. Colizza, R. Pastor-Satorras, and A. Vespignani. Reaction diffusion processes and metapopulation models in heterogeneous networks. *Nat. Phys.*, 3(4):276–282, 2007.
11. E. Côme and P. Latouche. Model selection and clustering in stochastic block models with the exact integrated complete data likelihood. Pre-print, *arXiv:1303.2962*, Mar. 2013.
12. A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.*, 107(6):65701, 2011.
13. S. Fortunato. Community detection in graphs. *Phys. Rep.*, 486(3):75–174, 2010.
14. R. Guimerà and M. Sales-Pardo. Missing and spurious interactions and the reconstruction of complex networks. *Proc. Natl. Acad. Sci. USA*, 106(52):22073–22078, 2009.
15. R. Guimerà and M. Sales-Pardo. A network inference method for large-scale unsupervised identification of novel drug-drug interactions. *PLOS Comput. Biol.*, 9(12):e1003374, Dec. 2013.
16. J. Hofman and C. Wiggins. Bayesian approach to network modularity. *Phys. Rev. Lett.*, 100(25):258701, 2008.
17. P. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.
18. B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83(1):016107, 2011.
19. D. B. Larremore, A. Clauset, and A. Z. Jacobs. Efficiently inferring community structure in bipartite networks. Pre-print, *arXiv:1403.2933*, Mar. 2014.
20. S. Merritt and A. Clauset. Scoring dynamics across professional team sports: tempo, balance and predictability. *EPJ Data Science*, 3:4, 2014.
21. M. E. J. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126, Feb. 2003.
22. M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford; New York, 2010.
23. A. O'Hagan. *Kendall's Advanced Theory of Statistics: Bayesian Inference. 2B*. Wiley, 1 edition, 2004.
24. T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, 2009.

25. R. K. Pan, K. Kaski, and S. Fortunato. World citation and collaboration networks: uncovering the role of geography in science. *Sci. Rep.*, 2:902, 2012.
26. Y. Park, C. Moore, and J. Bader. Dynamic networks from hierarchical bayesian graph clustering. *PLOS ONE*, 5(1):e8118, 2010.
27. L. Peel and A. Clauset. Detecting change points in the large-scale structure of evolving networks. Pre-print, *arXiv:1403.0989*, 2014.
28. T. Peixoto. Parsimonious module inference in large networks. *Phys. Rev. Lett.*, 110(14):148701, 2013.
29. T. Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X*, 4:011047, 2014.
30. M. A. Porter, P. J. Mucha, M. Newman, and C. M. Warmbrand. A network analysis of committees in the United States House of Representatives. *Proc. Natl. Acad. Sci. USA*, 102(20):7057–7062, 2005.
31. C. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, New York, 2007.
32. STATS LCC. Copyright 2014, 2014.
33. A. C. Thomas and J. K. Blitzstein. Valued ties tell fewer lies: Why not to dichotomize network edges with thresholds. Pre-print, *arXiv:1101.0788*, 2011.
34. Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *J. Am. Stat. Assoc.*, 82(397):8–19, 1987.
35. X. Yan, J. E. Jensen, F. Krzakala, C. Moore, C. R. Shalizi, L. Zdeborová, P. Zhang, and Y. Zhu. Model selection for degree-corrected block models. Pre-print, *arXiv:1207.3994*, 2012.
36. J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Explor. Artif. Intell. Millenn.*, 8:236–239, 2003.
37. P. Zhang, F. Krzakala, J. Reichardt, and L. Zdeborová. Comparative study for inference of hidden classes in stochastic block models. *J. Stat. Mech.*, 2012(12):P12021, 2012.