UNIVERSITY OF CAMBRIDGE
Department of Engineering

Inferring community characteristics in labelled networks

Author Name: Lawrence Tray

Supervisor: Ioannis Kontoyiannis

Date: 02/06/21

I hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Signed: *Lawrence Tray* *date:* 02/06/21

Technical Abstract

Labelled networks form a very common and important class of data, naturally appearing in numerous applications in science and engineering. A typical inference goal is to determine how the vertex labels (or *features*) affect the network's graph structure.

A standard approach has been to partition the network into blocks grouped by distinct values of the feature of interest. A block-based random graph model – typically a variant of the stochastic block model – is then used to test for evidence of asymmetric behaviour within these feature-based communities. Nevertheless, the resulting communities often do not produce a natural partition of the graph.

In this work, we introduce a new generative model, the feature-first block model (FFBM), which is more effective at describing vertex-labelled undirected graphs and also facilitates the use of richer queries on labelled networks. We develop a Bayesian framework for inference with this model, and we present a method to efficiently sample from the posterior distribution of the FFBM parameters. The FFBM's structure is kept deliberately simple to retain easy interpretability of the parameter values.

We apply the proposed methods to a variety of network data to extract the most important features along which the vertices are partitioned. The main advantages of the proposed approach are that the whole feature-space is used automatically and that features can be rank-ordered implicitly according to impact. Any features that do not significantly impact the high-level structure can be discarded to reduce the problem dimension. In cases where the vertex features available do not readily explain the community structure in the resulting network, the approach detects this and is protected against over-fitting. Results on several real-world datasets illustrate the performance of the proposed methods.

Nomenclature

We provide a list of frequently used symbols for reference.

Table 1: Nomenclature

Symbol	Meaning
α	MH acceptance probability
π	MCMC target density
ϕ	Softmax activation function
ψ	SBM parameters excluding b
θ	Feature-to-block generator parameters
ξ	Injected noise term in MALA
A	Graph adjacency matrix
b	Block membership vector
B	Total number of blocks
c	Dimensionality reduction cutoff
D	Feature-space dimension
e	Inter-block edge count matrix
E	Total number of edges
f	Fraction of vertices kept for training set
k	Degree sequence or standard deviation multiplier
$\mathcal{L}$	Average cross-entropy loss
N	Number of nodes in graph
q	MH proposal distribution
S	SBM description length
T	Length of Markov chain
$\mathcal{T}$	Retained sample set
U	Negative log-posterior for θ
w	Softmax weight vector
x	Feature vector
X	Feature matrix
y	Block membership posterior

Contents

1 Introduction

An important characteristic of many real-world networks is that they exhibit strong community structure, with most nodes often belonging to a densely connected cluster. Finding ways to recover the latent communities from the observed graph is an important task in many applications, including graph/network compression [1] and link prediction in incomplete networks [4].

In this work, we restrict our attention to vertex-labelled networks, and we refer to the vertex labels as *features*. A common goal is to determine whether a given feature impacts graphical structure. To answer this from a Bayesian perspective requires the use of a random graph model; the standard is the stochastic block model (SBM) [8]. This is a latent variable model where each vertex belongs to a single block and the probability two nodes are connected depends only on the block memberships of each. Numerous variants of this model have been considered – the most popular ones being the mixed-membership stochastic block model (MMSBM) [2] and the overlapping stochastic block model (OSBM) [19]. Effectively, these extend the model to allow each vertex to belong to multiple blocks simultaneously. However, one drawback of these graphical models as applied to labelled networks is that they do not automatically include vertex features in the random graph generation process. Approaches based on graph neural networks [7] that utilise vertex features have been developed but these lack the easy interpretability of the simpler models.

To analyse a labelled network using one of the simple SBM variants, a typical inference procedure would be to first partition the graph into blocks grouped by distinct values of the feature of interest, and then use the associated model to test for evidence of heterogeneous connectivity between the feature-grouped blocks. Nevertheless, this approach is limited as it can only consider one feature at a time. This makes it difficult to rank-order the features by magnitude of impact. Lastly, the feature-grouped blocks are often an unnatural partition of the graph, leading to a poor model fit. We would instead prefer to partition the graph into its most natural blocks and then find which of the available features – if any – best predict the resulting partition.

Thus motivated, we present a novel framework for modelling labelled networks, which we call the feature-first block model (FFBM). This is an extension of the SBM to labelled networks. We go on to present an efficient algorithm for sampling from the FFBM parameters, and to describe how the sampled parameters can be interpreted to determine which features have the largest impact on overall graphical structure.

2 Preliminaries

We first need a model for community-like structure in a network. For this we adopt the widely-used stochastic block model (SBM). Each node in the graph belongs to a unique community called a block. The probability that two nodes are connected depends only on the block memberships of each. Specifically, we will use the microcanonical variant of the SBM, proposed by Peixoto [14]. To allow for degree-variability between members of the same block, we adopt the degree-corrected formulation (DC-SBM), defined in (2.1).

For each integer $K \geq 1$, we use the notation $[K] := \{1, 2, \ldots, K\}$.

Definition 2.1 (Microcanonical DC-SBM) *Let $N \geq 1$ denote the number of vertices in an undirected graph. The block memberships are encoded by a vector $b \in [B]^N$, where B is the number of non-empty blocks. Let $e = (e_{rs})$ be the $N \times N$ symmetric matrix of edge counts between blocks, such that e_{rs} is the number of edges from block r to block s – or twice that number if $r = s$. Let $k = (k_i)$ denote the vector of length N containing the degree sequence of the graph, with k_i being the degree of vertex i.*

The graph's adjacency matrix $A \in \{0, 1\}^{N \times N}$ is generated by placing edges uniformly at random, but respecting the constraints imposed by e, b and k (hence the qualification 'microcanonical'). Specifically, A must satisfy the following, for all $r, s \in [B]$ and all $i \in [N]$:

$$e_{rs} = \sum_{i,j \in [N]} A_{ij} \mathbb{1}\{b_i = r\} \mathbb{1}\{b_j = s\} \qquad \text{and} \qquad k_i = \sum_{j \in [N]} A_{ij}. \tag{1}$$

We use the following notation to indicate this distribution:

$$A \sim \text{DC-SBM}_{\text{MC}}(b, e, k). \tag{2}$$

Appendix **??** describes how to compute the likelihood, $p(A|b, e, k)$, for graphs generated from the microcanonical DC-SBM.

3 Feature-first block model

In this section we propose a novel generative model for labelled networks. We call this the feature-first block model (FFBM).

Let N denote the number of nodes, B the number of blocks in the graph, and $\mathcal{X}$ the set of values each feature can take. We define the vector $x_i \in \mathcal{X}^D$ as the feature vector of vertex i, where D is the number of features associated with each vertex. For the datasets we analyse, we deal with binary feature flags so $\mathcal{X} = \{0, 1\}$. We write X for the $N \times D$ *feature matrix* containing the feature vectors $\{x_i\}_{i=1}^N$ as its rows.

For the FFBM, we start with the feature matrix X and generate a random vector of block memberships $b \in [B]^N$. For each vertex i, the block membership $b_i \in [B]$ is generated based on the feature vector x_i, independently between vertices. The conditional distribution of b_i given x_i also depends on a collection of weight vectors $\theta = \{w_k\}_{k=1}^B$, where each w_k has dimension D. Specifically, the distribution of b given X and θ is,

$$p(b|X, \theta) = \prod_{i=1}^{N} p(b_i|x_i, \theta) = \prod_{i=1}^{N} \phi_{b_i}(x_i; \theta) = \prod_{i=1}^{N} \frac{\exp\left(w_{b_i}^T x_i\right)}{\sum_{k=1}^{B} \exp\left(w_k^T x_i\right)}. \tag{3}$$

ϕ_{b_i} is the form of a softmax activation function; we deliberately exclude a bias term to ensure that we only leverage feature information. We will later also find it convenient to write the parameters θ as a $B \times D$ matrix of weights W.

More complex models based on different choices for the distributions ϕ_{b_i} above are also possible, but then deriving meaning from the inferred parameter distributions is more difficult.

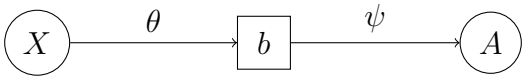


Figure 1: The feature-first block model (FFBM)

Once the block memberships b have been generated, we then draw the graph A from the microcanonical DC-SBM with additional parameters $\psi = \{\psi_e, \psi_k\}$:

$$A \sim \text{DC-SBM}_{\text{MC}}(b, \psi_e, \psi_k). \tag{4}$$

3.1 Prior selection

To complete the description of our Bayesian framework, priors on θ and ψ must also be specified. We place a Gaussian prior on θ with zero mean and covariance matrix $\sigma_\theta^2 I$, so that each element of θ has an independent $\mathcal{N}(0, \sigma_\theta^2)$ prior, with hyperparameter σ_θ^2:

$$p(\theta) \sim \mathcal{N}\left(\theta; 0, \sigma_\theta^2 I\right). \tag{5}$$

Interestingly, this choice of prior leads to a particularly simple distribution on the block membership vector b given X. We show in Appendix B.1 that, after integrating out θ, b is uniformly distributed given X:

$$p(b|X) = \int p(b|X,\theta)p(\theta)d\theta = B^{-N}. \tag{6}$$

This is an important simplification as evaluating $p(b|X)$ does not require an expensive Monte Carlo integration over θ nor does it require the exact value of X. Peixoto [14] proposes careful choices for the priors on the additional microcanonical SBM parameters ψ, which we adopt. The idea is to write the joint distribution on (b, e, k) as a product of conditionals, $p(b, e, k) = p(b)p(e|b)p(k|e, b) = p(b)p(\psi|b)$. In our case, conditioning on X is also necessary, leading to,

$$p(b, \psi|X) = p(b|X)p(\psi|b, X) = p(b|X)p(\psi|b), \tag{7}$$

where we used the fact ψ and X are conditionally independent given b. The exact form of the prior $p(\psi|b)$ is described in Appendix A.2. All that concerns the main argument is that it has an easily computable form.

4 Inference

Having completed the definition of the FFBM, we wish to leverage it to perform inference. Given a vertex-labelled graph (A, X), the goal is to sample from the posterior distribution of θ:

$$\theta \sim p(\theta|A, X). \tag{8}$$

We propose an iterative approach to obtain samples $\theta^{(t)}$ from $p(\theta|A, X)$. We first draw samples $b^{(t)}$ from the block membership posterior distribution, and then use each $b^{(t)}$ to obtain a corresponding sample $\theta^{(t)}$:

$$b^{(t)} \sim p(b|A, X), \tag{9}$$

$$\theta^{(t)} \sim p(\theta|X, b^{(t)}). \tag{10}$$

Both of these sampling steps can be implemented using Markov chain Monte Carlo via a Metropolis-Hastings algorithm [5], as a two-level Markov chain. See Appendix A.3 for more details on this.

As we show in the following section, the resulting samples for $\theta^{(t)}$ are unbiased in the sense that the expectation of their distribution is the target posterior distribution:

$$\mathbb{E}_{b^{(t)}}\left[p\left(\theta|X, b^{(t)}\right)\right] = \sum_{b\in[B]^N} p(\theta|X, b)p(b|A, X) = \sum_{b\in[B]^N} p(\theta, b|A, X) = p(\theta|A, X). \tag{11}$$

This is an example of a pseudo-marginal approach. Indeed, Andrieu and Roberts [3] show that (11) is sufficient to prove that, for large enough t, $\theta^{(t)} \sim \mathbb{E}_{b^{(i)}}\left[p(\theta|X, b^{(t)})\right] = p(\theta|A, X)$, as required.

The reason for splitting the Markov chain into two levels is that the summation over all latent states $b \in [B]^N$ required to directly compute the likelihood, $p(A|X, \theta) = \sum_{b\in[B]^N} p(A|b)P(b|X, \theta)$, is intractable, as the sum involves B^N terms. Figure 2 shows an overview of the proposed method. The Metropolis-Hastings accept/reject probabilities for the proposed b and θ samples are denoted α_b and α_θ, respectively.

Note the importance of the simplification in (6). As $p(b|X)$ in fact does not depend on X, knowing X is not necessary in order to obtain samples of the block membership vector b. And on the other level, in order to obtain samples of the parameter vector θ we use only b but not A, as θ and A are conditionally independent given b.

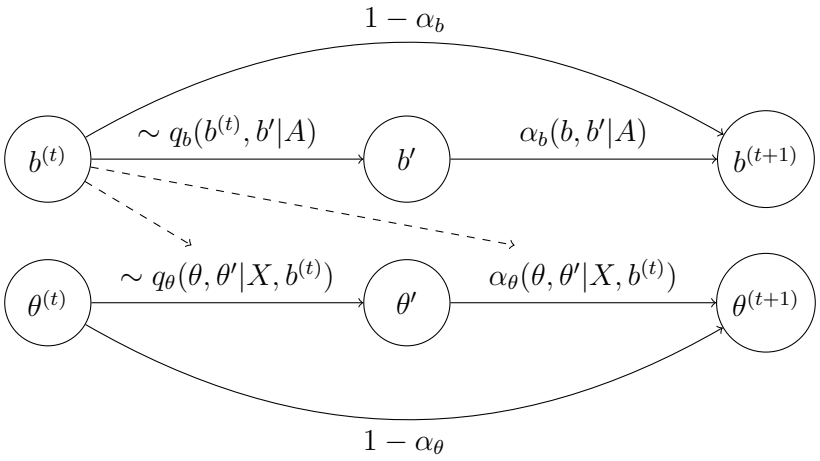


Figure 2: Sampling sequence.

4.1 Sampling block memberships

We adopt the Markov chain Monte Carlo procedure of [11], which relies on writing the posterior in the following form:

$$p(b|A, X) \propto p(A|b, X) \cdot p(b|X) = \pi_b(b). \tag{12}$$

Now $\pi_b(\cdot)$ is the un-normalised target density, which can be expressed as:

$$\pi_b(b) = p(b|X) \sum\nolimits_{\psi} p(A, \psi|b, X) = p(b|X)p(A, \psi^*|b, X) = p(A|b, \psi^*) \cdot p(\psi^*|b) \cdot p(b|X). \tag{13}$$

Since we are using the microcanonical SBM formulation, there is only one value of ψ that is compatible with the given (A, b) pair; recall the constraints in (1). We denote this value $\psi^* = \{\psi_k^*, \psi_e^*\}$. Therefore, the summation over all ψ reduces to just the single ψ^* term. We also define the microcanonical entropy of the configuration as,

$$S(b) = -\log \pi_b(b) = -\Big(\log p(A|b, \psi^*) + \log p(\psi^*, b|X)\Big). \tag{14}$$

This entropy can be thought of as the optimal "description length" of the graph. The exact form of the proposal q_b is explored thoroughly in [11] and not repeated here. There is a widely used library for Python made available under LGPL called `graph-tool` [13], which implements this algorithm. The only modification we make is in the block membership prior $p(b)$ that we replace with $p(b|X) = B^{-N}$, which cancels out in the MH accept-reject step as it is independent of b.

4.2 Sampling feature-to-block generator parameters

The invariant distribution we wish to target for the θ samples is the posterior of θ given the values of the pair (X, b). We write this as,

$$\pi_\theta(\theta) \propto p(\theta|X, b) \propto p(b|X, \theta)p(\theta) \propto \exp\left(-U(\theta)\right), \tag{15}$$

where we write $U(\theta)$ for the negative log-posterior. Let $y_{ij} \coloneqq \mathbb{1}\{b_i = j\}$ and $a_{ij} \coloneqq \phi_j(x_i; \theta)$. Discarding constant terms, we can write $U(\theta)$ as,

$$U(\theta) = \left(\sum_{i\in[N]}\sum_{j\in[B]} y_{ij} \log\frac{1}{a_{ij}}\right) + \frac{1}{2\sigma_\theta^2}\|\theta\|^2 = N \cdot \mathcal{L}(\theta) + \frac{1}{2\sigma_\theta^2}\|\theta\|^2; \tag{16}$$

See Appendix B.2 for the derivation. The function $U(\theta)$ is a typical objective function for neural network training. The first term is introduced by the likelihood. We collect it into $N \cdot \mathcal{L}(\theta)$, which is the cross-entropy between the graph-predicted and feature-predicted block memberships summed over all vertices. The second, introduced by the prior, brings a form of regularisation, guarding against over-fitting. Unlike in many applications where the goal is to find the minimiser of $U(\theta)$, our goal is to draw samples from the posterior $\pi_\theta(\cdot) \propto \exp(-U(\cdot))$. We can use ∇U as a useful heuristic to bias our proposal towards regions of higher target density. We therefore adopt the Metropolis-adjusted Langevin algorithm (MALA) [16]. Given the current sample θ, we generate a new sample θ' as,

$$\theta' = \theta - h\nabla U(\theta) + \sqrt{2h} \cdot \xi,$$

where $\xi \sim \mathcal{N}(0, I)$ and h is a step-size parameter which may vary with the sample index (see Appendix A.4). This leads to the proposal distribution,

$$q_\theta(\theta, \theta') \sim \mathcal{N}\left(\theta'; \theta - h\nabla U(\theta), 2hI\right).$$

Without the injected noise term, MALA is equivalent to gradient descent. We require the noise term ξ to fully explore the parameter space. The term ∇U has an easy to compute analytic form (derived in Appendix B.3). By noting that $\theta = \{w_k\}_{k=1}^B$, we write the derivative

with respect to each w_k as,

$$\frac{\partial U}{\partial w_k} = -\left(\sum_{i=1}^{N}\left\{\tilde{x}_i(y_{ik} - a_{ik})\right\} - \frac{w_k}{\sigma_\theta^2}\right). \tag{17}$$

After a proposed move is generated from q_θ, it is either accepted or rejected in the standard Metropolis-Hastings accept/reject step.

4.3 Sampling sequence

Up to this point, each $\theta^{(t)}$ update uses its corresponding $b^{(t)}$ sample. This means that the evaluation of $U(\theta)$ and $\nabla U(\theta)$ has high variance. This may lead to longer burn-in for the resulting Markov chain. The only link between $b^{(t)}$ and $\theta^{(t)}$ is in the evaluation of $U(\theta)$ and $\nabla U(\theta)$ which depends only on the matrix $y^{(t)}$ with entries $y_{ij}^{(t)} \coloneqq \mathbb{1}\{b_i^{(t)} = j\}$. We would rather deal with the expectation of each $y_{ij}^{(t)}$:

$$\mathbb{E}\left[y_{ij}^{(t)}\right] = \mathbb{E}_{b^{(t)}}\left[\mathbb{1}(b_i^{(t)} = j)\right] = p(b_i = j|A, X). \tag{18}$$

We can obtain an unbiased estimate for this quantity using the thinned b-samples after burn-in; the length of the burn-in and the details of the thinning are described in Appendix A.5. Let $\mathcal{T}_b$ denote the retained set of indices for the b-samples and $\mathcal{T}_\theta$ similarly for the θ-chain. The unbiased estimate for $y_{ij}^{(t)}$ using the restricted sample set $\mathcal{T}_b$ is denoted $\hat{y}_{ij}$:

$$\hat{y}_{ij} \coloneqq \frac{1}{|\mathcal{T}_b|}\sum_{t\in\mathcal{T}_b} y_{ij}^{(t)} = \frac{1}{|\mathcal{T}_b|}\sum_{t\in\mathcal{T}_b}\mathbb{1}\{b_i^{(t)} = j\}. \tag{19}$$

The same matrix $\hat{y}$ is used in the update step for each $\theta^{(t)}$. This way, it is not necessary to run the b and θ Markov chains concurrently. Instead, we run the b-chain to completion and use it to generate $\hat{y}$. This affords us the flexibility to vary the lengths of the b and θ-chains. Furthermore, the changeover from $y^{(t)}$ to $\hat{y}$ reduces the burn-in time for the θ-chain by reducing the variance in the evaluation of U and ∇U. A detailed description of the overall algorithms is given in Appendix C.1.

4.4 Dimensionality reduction

Once the samples $\left\{\theta^{(t)}\right\} \sim p(\theta|A, X)$ have been obtained, we can compute the empirical mean and standard deviation of each component of θ. Switching back to matrix notation

we define $\theta = W$, such that W_{ij} is the weight component for block i and feature j, we can define:

$$\hat{\mu}_{ij} \coloneqq \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} W_{ij}^{(t)} \qquad \text{and} \qquad \hat{\sigma}_{ij}^2 \coloneqq \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} \left(W_{ij}^{(t)} - \hat{\mu}_{ij} \right)^2 \tag{20}$$

A simple heuristic to discard the least important features requires specifying a cutoff $c > 0$ and a multiplier $k > 0$. We define the function $\mathcal{F}_i(j)$ as in (21) and only keep features with indices $d \in \mathcal{D}'$, where $\mathcal{D}'$ is given in (22).

$$\mathcal{F}_i(j) \coloneqq (\hat{\mu}_{ij} - k\hat{\sigma}_{ij}, \hat{\mu}_{ij} + k\hat{\sigma}_{ij}) \cap (-c, +c), \tag{21}$$

$$\mathcal{D}' \coloneqq \{j \in [D] : \exists i \in [B] \text{ s.t. } \mathcal{F}_i(j) = \emptyset\} \tag{22}$$

Intuitively, this means discarding any feature j for which $\hat{\mu}_{ij} \pm k\hat{\sigma}_{ij}$ lies outside the region $(-c, c)$ for all block indices i. If we were to use the Laplace approximation for the posterior $p(W_{ij}|A, X) \approx \mathcal{N}(W_{ij}; \hat{\mu}_{ij}, \hat{\sigma}_{ij}^2)$, then this is analogous to a hypothesis test on the value of W_{ij} as in (23); then $\mathcal{D}'$ comprises all features i for which H_1 is not rejected at least once for some $j \in [B]$

$$H_0 : |W_{ij}| \leq c \qquad H_1 : |W_{ij}| > c \tag{23}$$

The multiplier k in equation 21 determines the degree of significance of the result. However, as the Laplace approximation is not exact we will only treat this dimensionality reduction method as a useful heuristic and not an exact method. Conversely, we could fix $k = k_0$ and the dimension of our reduced feature set $|\mathcal{D}'| = D'$. We would then like to find the largest value of c such that $|\mathcal{D}'| = D'$ given $k = k_0$. This is summarised in equation 24.

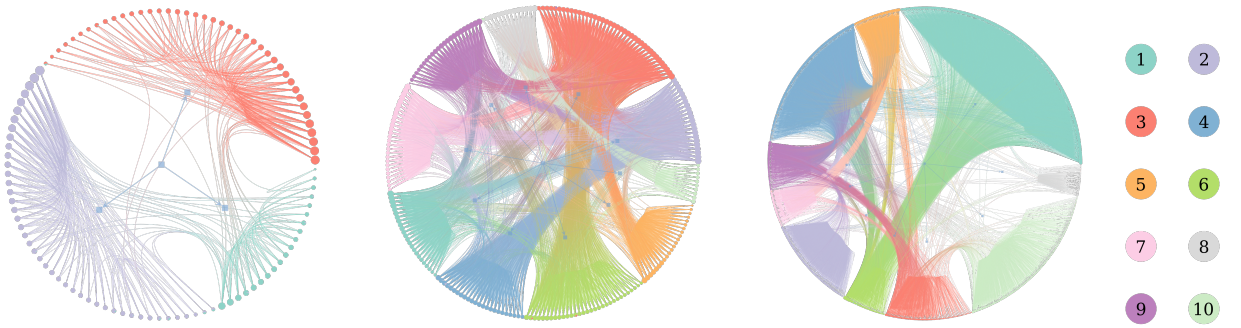
$$c^* = \arg\max\{c > 0 \ : \ |\mathcal{D}'| = D', k = k_0\}. \tag{24}$$

For an algorithmic implementation of this method refer to appendix C.1. This approach has the advantage that it fixes the number of reduced dimensions.

5 Experimental results

We apply the developed methods to a variety of datasets. These are chosen to span a range of node counts N, edge counts E and feature space dimension D. We consider the following:

- **Political books** [9] ($N = 105, E = 441, D = 3$) – network of Amazon book sales about U.S. politics, published close to the presidential election in 2004. Two books are connected if they were frequently co-purchased by customers. Vertex features encode the political affiliation of the author (liberal, conservative, or neutral).
- **Primary school dynamic contacts** [17] ($N = 238, E = 5539, D = 13$) – network of face-to-face contacts amongst students and teachers at a primary school in Lyon, France. Two nodes are connected if the two parties shared a face-to-face interaction over the school-day. Vertex features include class membership (one of 10 values: 1A-5B), gender (male, female) and teacher status encoded as an 11th school-class. No further identifiable information is retained. We choose to analyse just the second day of results.
- **Facebook egonet** [6] ($N = 747, E = 30025, D = 480$) – an assortment of Facebook users' friends lists. Vertex features are extracted from each user's profile and are fully anonymised. They include information about education history, languages spoken, gender, home-town, birthday etc. We focus on the egonet with id 1912.



(a) Polbooks (b) School (c) Facebook egonet (d) Legend

Figure 3: Networks laid out and coloured according by inferred block memberships $\hat{y}$ for a given experiment iteration. Visualisation performed using *graph-tool* [13].

We require metrics to assess performance. This can be split into two separate components: the microcanonical SBM fit (concerned with the b-samples) and the fit of the feature-to-block generator (concerned with the θ-samples).

Table 2: Experimental results averaged over $n = 10$ iterations (mean $\pm$ standard deviation).

Dataset	B	D	D'	$\bar{S}_e$	$\bar{\mathcal{L}}_0$	$\bar{\mathcal{L}}_1$	c^*	$\bar{\mathcal{L}}'_0$	$\bar{\mathcal{L}}'_1$
Polbooks	3	3	–	2.250 ± 0.000	0.563 ± 0.042	0.595 ± 0.089	–	–	–
School	10	13	10	1.894 ± 0.004	0.787 ± 0.127	0.885 ± 0.129	1.198 ± 0.249	0.793 ± 0.132	0.853 ± 0.132
FB egonet	10	480	10	1.626 ± 0.003	1.326 ± 0.043	1.538 ± 0.069	0.94 ± 0.019	1.580 ± 0.150	1.605 ± 0.106

Starting with the SBM, recall that the quantity $S(b)$ in (14) can be interpreted as an ideal "description length" of the partition imposed by b. We define a simple metric $\bar{S}_e$ to gauge the fit of the SBM, as the description length per entity (i.e., divided by the total number $N + E$ of nodes plus edges), averaged over the b-samples:

$$\bar{S}_e \coloneqq \frac{1}{(N+E)|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} S\left(b^{(t)}\right). \tag{25}$$

Next, to assess the performance of the feature-to-block predictor, we partition the vertex set $[N]$ into training and test sets. We choose to randomly partition the vertices on each experiment run, so that a constant fraction f of the available vertices go to form the training set $\mathcal{G}_0$ and the remainder are held out to form the test set $\mathcal{G}_1$. The b-chain is run using the whole network but we only use vertices $v \in \mathcal{G}_0$ to train the θ-chain. Because $|\mathcal{G}_0| \neq |\mathcal{G}_1|$ in general, we cannot use the un-normalised log-target U from (16) for comparison, as the total cross-entropy loss scales with the size of each data set but the prior term stays constant. We therefore use the average cross-entropy loss over each set,

$$\bar{\mathcal{L}}_\star \coloneqq \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} \mathcal{L}_\star\left(\theta^{(t)}\right), \quad \text{where} \quad \mathcal{L}_\star\left(\theta^{(t)}\right) \coloneqq \frac{1}{|\mathcal{G}_\star|} \sum_{i \in \mathcal{G}_\star} \sum_{j \in [B]} \hat{y}_{ij} \log \frac{1}{\phi_j\left(x_i; \theta^{(t)}\right)}, \tag{26}$$

where $\star \in \{0, 1\}$ indicates whether the training or test set is being considered.

Table 2 summarises the results for each experiment; a complete list of all the hyperparameters used is given in Appendix C.2. We also apply the dimensionality reduction method of Section 4.4 to the two higher dimensional datasets (the school and FB egonet). For this we use equation (24) with $k = 1$, in order to reduce the dimension from D to a desired D'. We then retrain the feature-block predictor using only the retained feature set $\mathcal{D}'$, and report the log-loss over the training and test sets for the reduced classifier – denoted $\bar{\mathcal{L}}'_0$ and $\bar{\mathcal{L}}'_1$ respectively. These values are also shown in Table 2.

Based on these results, some remarks are in order. Firstly, the variance of the test loss $\bar{\mathcal{L}}_1$ tends to be higher than the training loss $\bar{\mathcal{L}}_0$. This is expected, as the test set is smaller than the training set and hence more susceptible to variability in its construction. Indeed, most

of the variance in the evaluation of $\bar{\mathcal{L}}_0$ and $\bar{\mathcal{L}}_1$ comes from the random partitioning of the graph into training and test sets. Secondly, it can be seen that the dimensionality reduction procedure brings the training and test losses closer together. This implies that the features we keep are indeed correlated with the underlying graphical partition and that the approach generalises correctly.

The average description length per entity, $\bar{S}_e$, of the graph, has very small variance, suggesting that the detected communities can be found reliably (to within an arbitrary relabelling of blocks). For reference, we plot an inferred partition for each of the graphs in Figure 3. The polbooks graph yields the cleanest separation between blocks but nonetheless the inferred partitions for the other datasets do succeed at dividing the graph into densely connected clusters.

Nevertheless, the cross-entropy loss over the whole training set may be too coarse a measure of model fit. It is often the case that we have good feature-based explanations for some but not all of the detected blocks. We wish to define a new measure of fit specific to each detected block. We could use a binary cross-entropy function but this would necessarily use the whole test/training set and so few vertices will belong to the block of interest. We instead prefer to compute quantities over the set of all vertices associated with block j, defined as,

$$\mathcal{B}_\star(j) \coloneqq \{i \in \mathcal{G}_\star : \hat{b}_i = j\} \quad \text{where} \quad \hat{b}_i \coloneqq \arg\max_j \hat{y}_{ij}. \tag{27}$$

Recall that $\hat{y}_{ij}$ is our estimate for the block membership posterior (19) using only information from the adjacency matrix A. Again, $\star \in \{0, 1\}$ toggles between training and test sets. Now $\mathcal{B}_\star(j)$ is the set of all vertices within the training or test set that have maximum a posteriori probability of belonging to set j. We choose to resolve ties consistently by picking the lower index. We now define the accuracy for block j as,

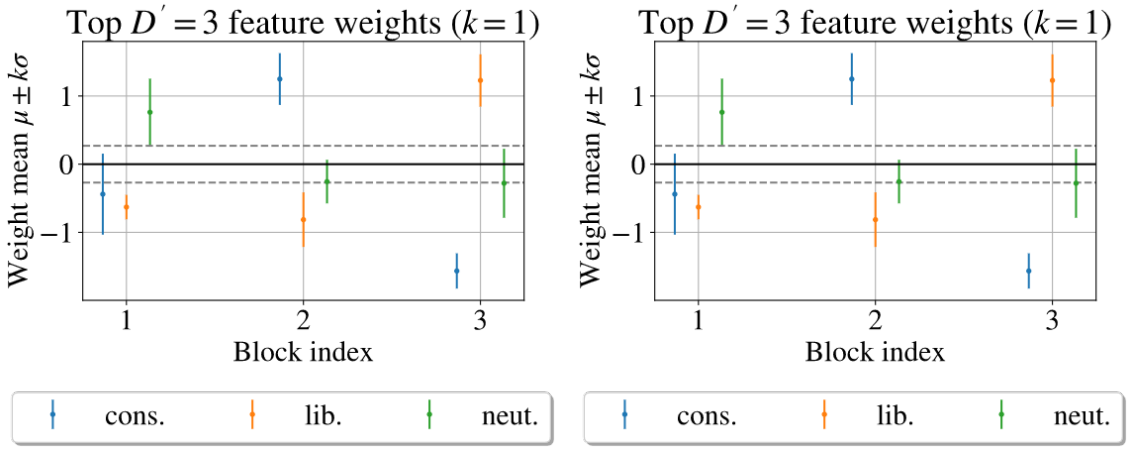
$$\eta_\star(j) \coloneqq \frac{1}{|\mathcal{B}_\star(j)| \cdot |\mathcal{T}_\theta|} \sum_{i \in \mathcal{B}_\star(j)} \sum_{t \in \mathcal{T}_\theta} \mathbb{1}\left\{\hat{b}_i = \arg\max_j \phi_j\left(x_i; \theta^{(t)}\right)\right\}. \tag{28}$$

This is effectively testing whether the feature-to-block predictions and the graph-based predictions agree in their largest component. We call this metric, $\eta_\star(j)$, the *block-accuracy* for block for j.

For each of the experiments, we plot the collected θ-samples for the features that survive the dimensionality reduction procedure. We also plot the per-block accuracy for the original-dimension classifiers.

5.1 Political books

The goal here is to determine whether the authors' political affiliations are a good predictor of the overall network structure. We choose to partition the network into $B = 3$ communities as we only have this many distinct values for political affiliation (conservative, liberal or neutral). From Figure 4a we see that all 3 blocks have a distinct political affiliation as their



(a) θ-samples

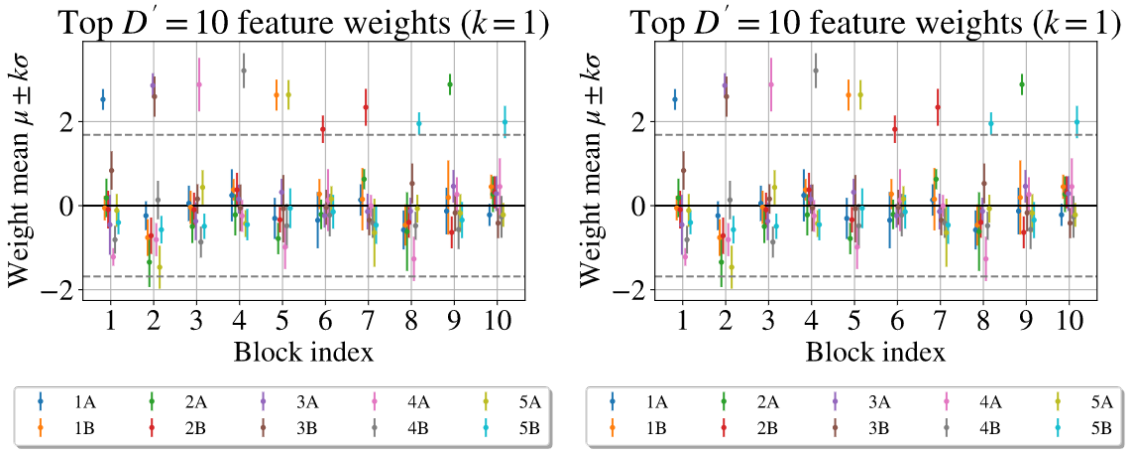
(b) Per-block accuracy $\eta(j)$

Figure 4: Political books dataset

largest positive component. This strongly indicates that political affiliation is indeed the axis which best predicts the 3-way natural partition of the graph into blocks. Furthermore, in Table 2 we see that the training and test losses are very similar and both are low in magnitude. This provides further evidence to the claim that political affiliation is a very appropriate explanatory variable for the overall network structure.

5.2 Primary school dynamic contacts

We choose the number of communities $B = 10$, in line with the total number of school classes. As before, we sample the block-generator parameters θ and employ the dimensionality reduction technique of Section 4.4, with standard deviation multiplier $k = 1$ to pick out the top $D' = 10$ features. We then plot the weights for the resulting features $d \in \mathcal{D}'$ in Figure 5a. It is rather obvious that only the pupils' class memberships (1A-5B) have remained significant; gender and teacher/student status have been discarded, meaning that these are not good predictors of overall macro-structure. The vast majority of blocks are composed of a single class. However, some blocks have two comparably good classes as their predictor.



(a) θ-samples

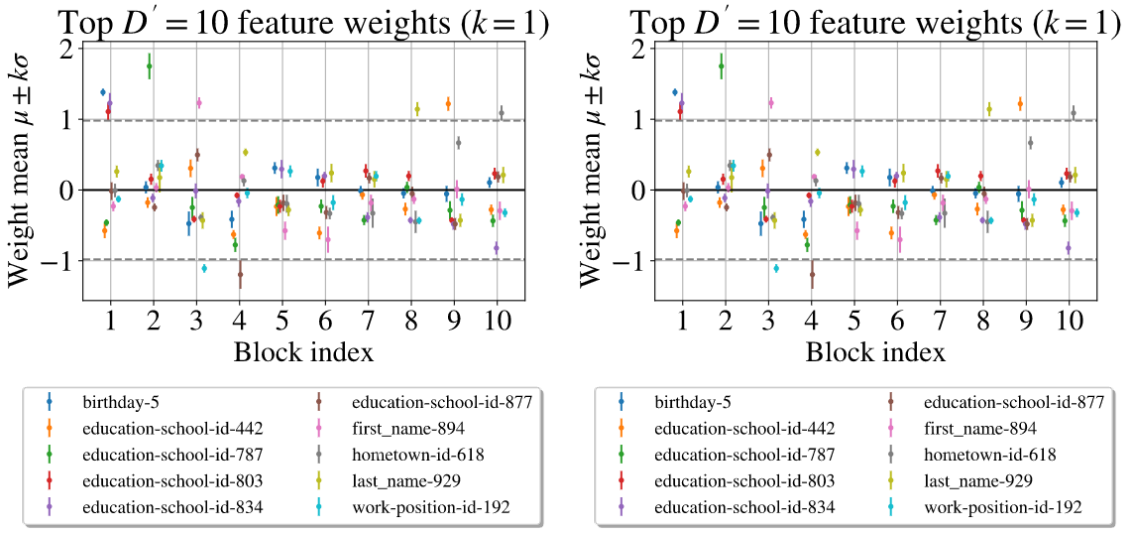
(b) Per-block accuracy $\eta(j)$

Figure 5: Primary school dynamic contacts dataset

For example, block 2 contains classes 3A and 3B as its 2 best predictors. This suggests that the social divide between classes is less pronounced for pupils in year 3. Conversely, some classes are found to extend over two detected blocks (class 2B spans blocks 6 and 7) but we nonetheless do not have a feature which explains the division. The most surprising block is number 5, which has comparable weightings for classes 5A and 1B. Perhaps there was a joint event between those two classes on the day the data were collected.

5.3 Facebook egonet

We choose $B = 10$ and $D' = 10$ for this experiment. The selected features (Figure 6a) are those that best explain the high-level community structure. The majority of them are education related. Nevertheless, for $D' = 10$ we only have good explanations for the makeup of some of the detected blocks; several blocks in Figure 6a do not have high-magnitude components for $D' = 10$. When the feature dimension is very large, it becomes increasingly likely that a particular feature may uniquely identify a small set of nodes. If these nodes are all part of the same community, then the classifier will overfit for that particular parameter. The regularisation term imposed by the prior goes some way towards alleviating this problem. Nevertheless, we see in Figure 6a that the feature `birthday-5` has a very high weight as it relates to block 1 – but it is highly unlikely that birthdays determine graphical structure.



(a) θ-samples

(b) Per-block accuracy $\eta(j)$

Figure 6: Facebook egonet dataset

6 Extensions

The greatest limitation of the current FFBM formulation is that it can only explain structure at the macro-scale. In other words, we cannot explain structure within each detected block. Future work will benefit from extending the FFBM to be hierarchical in nature. This would be a relatively natural extension. Indeed, the SBM has already been extended to a hierarchical form, often called the nested SBM [12]. The idea is to divide each block into sub-blocks and so on recursively until a specified depth is reached. The full block membership for a particular vertex now encodes the memberships at all levels of the hierarchy.

The necessary modification of the feature-to-block generator is also rather natural. Given the nested SBM, we would have a hierarchy of generators, each generating a block membership at a particular level of the hierarchy. Nevertheless, this does pose some practical issues for scalability; supposing we have L levels in our hierarchy and each divides the parent block into B sub-blocks, then the number of distinct generators necessary scales as $O(B^L)$. To avoid exponential growth in the number of model parameters, we could apply some form of dimensionality reduction as we descend the layers so that each generator is only given relevant features as input.

7 Conclusion

The Feature-First Block Model (FFBM) introduced in this paper is a new generative model for labelled networks, developed in order to address difficulties of other graphical models when testing how vertex features affect community structure. The idea is to divide the graph into its most natural partition and test whether the vertex features can accurately explain this partition. It is relatively easy to find vertex features that are in some way correlated with the graphical structure. Nonetheless, only when we find the feature that best describes the most pronounced partition do we have a stronger case for causation.

Using this new model, we go on to describe an efficient inference algorithm to sample the parameters θ of the feature-to-block generator. This takes the form of a two-level Markov chain, used to sample the block memberships b and block generator parameters θ. These chains can either be executed in parallel or the empirical mean of the b-samples can be used as the input to the θ-chain. The latter option reduces the variance in our evaluation of the target distribution and thus shortens burn-in.

The overall method is shown to be effective at extracting and describing the most natural communities in a labelled network. Nevertheless, the approach can only currently explain the structure at the macro-scale. We cannot explain structure within each block. Future work will benefit from extending the FFBM to be hierarchical in nature. That way, the structure of the network can be explained at all length-scales of interest. So long as data collection techniques remain ethical and care is taken to respect personal privacy, such empowered decision-making can only help humankind.

References

[1] Emmanuel Abbe. Graph compression: The effect of clusters. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, 2016. doi: 10.1109/ALLERTON.2016.7852203.

[2] Edo M Airoldi, David Blei, Stephen Fienberg, and Eric Xing. Mixed membership stochastic blockmodels. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2008/file/8613985ec49eb8f757ae6439e879bb2a-Paper.pdf.

[3] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697 – 725, 2009. doi: 10.1214/07-AOS574. URL https://doi.org/10.1214/07-AOS574.

[4] Solenne Gaucher, Olga Klopp, and Geneviève Robin. Outliers detection in networks with missing links, 2020.

[5] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444. URL http://www.jstor.org/stable/2334940.

[6] Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf.

[7] Nikhil Mehta, Lawrence Carin Duke, and Piyush Rai. Stochastic blockmodels meet graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4466–4474. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/mehta19a.html.

[8] Krzysztof Nowicki and Tom A. B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001. doi: 10.1198/016214501753208735. URL https://doi.org/10.1198/016214501753208735.

[9] Boris Pasternak and Ivor Ivask. Four unpublished letters. *Books Abroad*, 44(2):196–200, 1970. ISSN 00067431. URL http://www.jstor.org/stable/40124305.

[10] Tiago P. Peixoto. Parsimonious module inference in large networks. *Physical Review Letters*, 110(14), Apr 2013. ISSN 1079-7114. doi: 10.1103/physrevlett.110.148701. URL http://dx.doi.org/10.1103/PhysRevLett.110.148701.

[11] Tiago P. Peixoto. Efficient monte carlo and greedy heuristic for the inference of stochastic block models. *Physical Review E*, 89(1), Jan 2014. ISSN 1550-2376. doi: 10.1103/physreve.89.012804. URL http://dx.doi.org/10.1103/PhysRevE.89.012804.

[12] Tiago P. Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X*, 4:011047, Mar 2014. doi: 10.1103/PhysRevX.4.011047. URL https://link.aps.org/doi/10.1103/PhysRevX.4.011047.

[13] Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014. doi: 10.6084/m9.figshare.1164194. URL http://figshare.com/articles/graph_tool/1164194.

[14] Tiago P. Peixoto. Nonparametric bayesian inference of the microcanonical stochastic block model. *Physical Review E*, 95(1), Jan 2017. ISSN 2470-0053. doi: 10.1103/physreve.95.012317. URL http://dx.doi.org/10.1103/PhysRevE.95.012317.

[15] Jaakko Riihimäki and Aki Vehtari. Laplace approximation for logistic gaussian process density estimation and regression, 2013.

[16] Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341 – 363, 1996. doi: bj/1178291835. URL https://doi.org/.

[17] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, and Philippe Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):1–13, 08 2011. doi: 10.1371/journal.pone.0023176. URL https://doi.org/10.1371/journal.pone.0023176.

[18] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[19] Jun Zhu, Jiaming Song, and Bei Chen. Max-margin nonparametric latent feature models for link prediction, 2016.

Appendices

A Additional material

A.1 SBM likelihood and prior

For reference, we provide the likelihood and priors for the microcanonical SBM proposed by Peixoto [14]. We have that the graph A is drawn from the SBM,

$$A \sim \text{DC-SBM}_{\text{MC}}(b, e, k), \tag{29}$$

with edges placed uniformly at random but respecting the constraints imposed by b, e and k. The computation of the likelihood, $p(A|k, e, b)$, is simply a case of counting the number of configurations that yield the same adjacency matrix and dividing by the total number of configurations possible. This is what gives this formulation the 'microcanonical' moniker. If we consider the half-edges to be distinguishable for a moment, the total number of configurations that satisfy the e constraint is,

$$\Omega(e) = \frac{\prod_r e_r!}{\prod_{r,s:r<s} e_{rs}! \cdot \prod_r e_{rr}!!}, \tag{30}$$

where $e_r \coloneqq \sum_s e_{rs}$ and $(2m)!! \coloneqq 2^m m!$. Nevertheless, a great number of these $\Omega(e)$ total configurations yield the same graph A. We denote the number of configurations that yield the adjacency matrix A with,

$$\Xi(A) \coloneqq \frac{\prod_i k_i!}{\prod_{i,j:i<j} A_{ij}! \prod_i A_{ii}!!}. \tag{31}$$

Note the similarity between the forms of $\Omega(e)$ and $\Xi(A)$ as e is effectively the adjacency matrix of the block-graph. With these defined we can write the overall likelihood as the ratio of the two:

$$p(A|k, e, b) = \frac{\Xi(A)}{\Omega(e)}. \tag{32}$$

Obviously, this form is only defined if A respects the constraints imposed by (k, e, b) else the likelihood is 0.

A.2 SBM prior

We turn our attention to the prior on the SBM parameters, necessary for Bayesian analysis. As discussed in the main text, the block membership vector b is an intermediate variable in the FFBM so we are not free to choose a prior for it. Nevertheless, we can define the prior on (e, k) as a conditional on b. We switch back to the FFBM notation of $\psi = \{\psi_e, \psi_k\}$ to encapsulate e and k. We recall for reference the prior $p(\psi|b)$ from [14]:

$$p(\psi_e = e, \psi_k = k|b) = p(e|b)p(k|e, b) = \left[\left\{\begin{Bmatrix} B \\ 2 \end{Bmatrix}\right\}^{-1}\right] \cdot \left[\prod_r \frac{\prod_j \eta_j^r!}{n_r! q(e_r, n_r)}\right], \tag{33}$$

where $\left\{\begin{matrix} n \\ m \end{matrix}\right\}$ is shorthand for $\binom{n+m-1}{m} = \frac{(n+m-1)!}{(n-1)!(m)!}$, which can be thought of as the total number of distinct histograms produced by m samples in n bins. The value $E = \frac{1}{2}\sum_{r,s} e_{rs}$ is the total number of edges in the graph. Importantly, E is not allowed to vary and so $p(e|b)$ is uniform in e. The variable η_j^r denotes the number of vertices in block r that have degree j; formally, $\eta_j^r \coloneqq \sum_i \mathbb{1}\{b_i = r\}\mathbb{1}\{k_i = j\}$. The denominator $q(m, n)$ denotes the number of different histograms produced by m samples in at most n non-zero bins that sum to m. Finally, $e_r \coloneqq \sum_s e_{rs}$ is the total number of half edges in block r and $n_r \coloneqq \sum_i \mathbb{1}\{b_i = r\}$ is the number of vertices assigned to block r.

These priors were chosen carefully in [14] to more closely match the structure of empirical networks than simple uniform priors. We do not repeat these arguments here.

A.3 Metropolis-Hastings

Recall from Section 4, we state that we can implement sampling from the b and θ-chains through the Metropolis-Hastings algorithm [5]. For reference, we define the prerequisites for implementing the Metropolis-Hastings algorithm.

Suppose we wish to draw samples for some general x from the normalised density $\pi^*(x)$. We just need to define a proposal distribution $q(x, x')$ for proposing a move $x \to x'$ and be able to evaluate an un-normalised form of the target distribution, denoted $\pi(x) \propto \pi*(x)$, point-wise. Each proposed move $x \to x'$ is then accepted with probability α (34) else it is rejected and we stay at x.

$$\alpha(x, x') = \min\left(\frac{\pi(x')q(x', x)}{\pi(x)q(x, x')}, 1\right). \tag{34}$$

This accept-reject step ensures the resulting Markov Chain is in detailed balance with the target distribution $\pi(\cdot)$. Therefore, the set of samples $\{x^{(t)}\} \sim \pi^*(x)$

A.4 Choosing the MALA step-size

Recall that in Section 4.2 we used the Metropolis-adjusted Langevin algorithm (MALA) to sample from the θ-chain of the block membership generator parameters. At iteration t, the proposed sample is generated by:

$$\theta' = \theta^{(t)} - h_t \nabla U(\theta^{(t)}) + \sqrt{2h_t} \cdot \xi. \tag{35}$$

There are two competing objectives when choosing the step-size h_t. On the one hand, h_t needs to be large so that the sampler arrives at a high density region quickly, while too large a step-size would lead to low acceptance rates and thus inefficient sampling. An effective strategy is to use *simulated annealing*: allow h_t to slowly decrease with t, as long as $h_t > 0$ for all t and also:

$$\sum_{t=1}^{\infty} h_t = \infty, \qquad \text{and} \qquad \sum_{t=1}^{\infty} h_t^2 < \infty. \tag{36}$$

Following Welling and Teh [18], we adopt the polynomially decaying step-sizes, $h_t = \alpha(\beta+t)^{-\gamma}$, where $\alpha > 0$, $\beta > 0$ and $\gamma \in (1/2, 1]$ are hyper-parameters. We make the specific choices,

$$\alpha = \frac{250 \cdot s}{N}, \qquad \beta = 1000, \qquad \gamma = 0.8, \tag{37}$$

where N is the number of data-points and s, the *step-size scaling*, is the only free parameter.

As an aside, we note that an interesting modification to MALA can be made if we skip out the Metropolis-Hastings accept/reject step entirely. If this is done, the algorithm is instead called the stochastic gradient Langevin dynamics (SGLD) [18]. This speeds up computation at the expense of the exactness of the method. Indeed, it relies on the fact that as the step-size is annealed to 0, the acceptance probability gets close to 1. Figure 7 gives empirical backing to this approximate rule-of-thumb. Nevertheless, as SGLD is not exact, we only use MALA for experimentation.

In order to choose the exact value of the *step-size scaling* s for each experiment, we must quantify the trade-off between acceptance ratio and burn-in time. We illustrate the adopted method using the primary school dataset [17] as an example. We run the b-chain for $T_b = 1,000$ iterations and subsample such that $\mathcal{T}_b$ is computed with $\kappa_b = 0.2$ and $\lambda_b = 5$. We then use

this to run the θ-chain for $T_\theta = 10,000$ iterations. The acceptance ratio – which we denote r_α – is simply the fraction of proposed moves which we accept. However, to measure burn-in we define a new metric: the mean of the objective function averaged over all samples,

$$\bar{U} := \frac{1}{T_\theta} \sum_{t \in [T_\theta]} U\left(\theta^{(t)}\right). \tag{38}$$

Since the chain will equilibrate in the vicinity of a minima in $U(\theta)$, the average over all samples is a rough indicator of the speed of burn-in. The higher the value of $\bar{U}$, the longer we infer the chain took to burn in and reach equilibrium. This method relies on the assumption that the random starting point $\theta^{(0)}$ has a high value for the objective function $U(\theta^{(0)})$; this assumption holds as the domain of θ is a high dimensional space so we are unlikely to get a low $U(\theta)$ from a lucky guess. We plot the acceptance ratio r_α and average objective $\bar{U}$ for 10 values of the step-size multiplier s logarithmically spaced in the range $(10^{-2}, 10^1)$ on Figure 7.

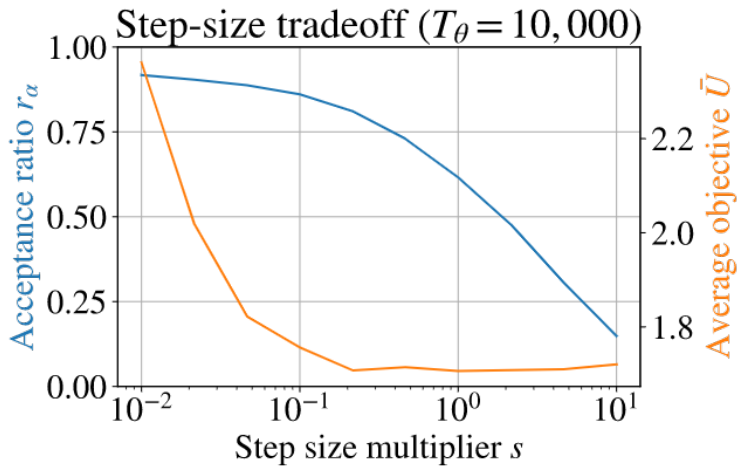


Figure 7: Primary school dataset [17] illustration of step-size tradeoff

We see immediately that the acceptance ratio r_α steadily decreases with s. This is expected as large jumps for each proposal make it unlikely we stay in a high density region of our target distribution. A low acceptance ratio leads to wasted samples and is obviously undesirable.

Nevertheless, $\bar{U}$ decreases with s indicating shorter burn-in times for larger step-sizes. However, this effect plateaus for $s > 0.2$ for this particular dataset. Therefore, we choose $s = 0.2$ as our step-size multiplier for the primary school dataset as this yields the best trade-off between r_α and $\bar{U}$. The step-sizes for the other datasets were chosen following the same argument but we do not repeat the process here.

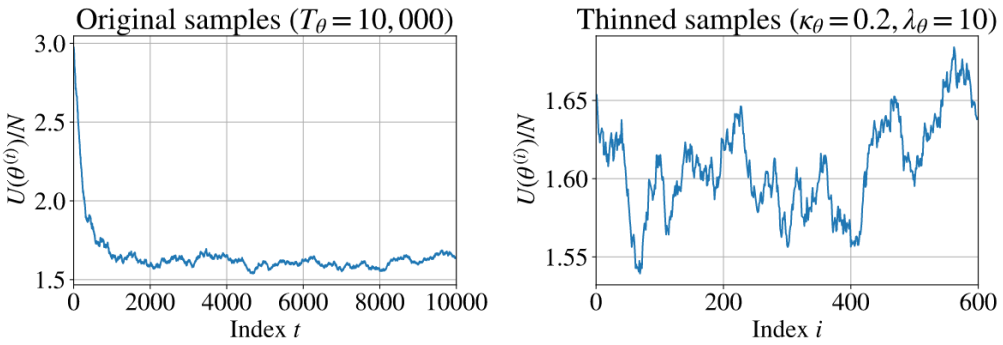
A.5 Burn-in and thinning

When sampling from the b and θ-chains described in Section 4, we respectively generate T_b and T_θ samples total. We discard an initial proportion $\kappa_\star \in (0, 1)$ of the samples as corresponding to a "burn-in" period required for the distribution of the chain to reach a distribution close to our target, and we also "thin" the remaining samples to obtain a less-dependent version. For $\star \in \{b, \theta\}$, the remaining sample sets are denoted $\mathcal{T}_\star$ in the notation of Section 4.3,

$$\mathcal{T}_\star = \{T_\star \kappa_\star + i\lambda_\star : 0 \leq i \leq \lfloor T_\star(1 - \kappa_\star)/\lambda_\star \rfloor\}, \tag{39}$$

where $\lambda_\star$ controls the thinning. The choice of $\kappa_\star$ can be determined by plotting the log-target (either $S(b^{(t)})$ or $U(\theta^{(t)})$ as a function of t, and choosing $\kappa_\star$ to encompass the region where the log-target has roughly reached equilibrium. As we do not leverage sample independence, $\lambda_\star$ can be chosen less rigorously; we often simply use $\lambda_b = 5$ and $\lambda_\theta = 10$.

By way of illustration, we can consider the primary school dataset [17]. We plot the normalised objective function $U\left(\theta^{(t)}\right)/N$ with respect to MALA iteration on Figure 8a – using the hyperparameters given in Appendix C.2. We see that the chain converges to the modal neighbourhood quickly (within about 2000 iterations of a total 10,000). Nevertheless, we pick $\kappa_\theta = 0.4 > 0.2$ to be on the safe-side. $\lambda_\theta = 10$ is chosen somewhat arbitrarily as we do not require neighbouring samples to be independent. However, this thinning also has the advantage of speeding up computation of quantities that require an average over all the retained samples – as $|\mathcal{T}_\theta| < T_\theta$.



(a) Original samples $t \in [T_\theta]$ (b) Thinned samples $t \in \mathcal{T}_\theta$ indexed by i

Figure 8: Primary school dataset normalised objective function against sample index

We plot the retained samples on Figure 8b. $U(\theta^{(i)})/N$ does bounce around a small amount but this is expected as we are sampling from the posterior rather than converging to the

MAP solution. There does appear to be variation over a long time-scale but this is acceptable as we do not strictly require sample independence.

A.6 Initializing the b-chain

For the purposes of the FFBM model, the number of blocks B is a constant which must be specified. If the choice of B is influenced by the observed data, then the analysis is no longer "fully Bayesian" and belongs to the class of methods referred to as "empirical Bayes." However, as the number of blocks only specifies the coarseness of the analysis, it is reasonable to allow it to vary. Indeed, Peixoto [10] shows that for a fixed average degree the maximum number of detectable blocks scales as $O(\sqrt{N})$ where N is the number of vertices.

If B is allowed to vary in the b-chain (i.e., when new blocks can be created and empty blocks are allowed), then the chain can be run until a minimum description length (MDL) solution is reached. We take the number of non-empty blocks in the MDL solution to be our fixed block number B for subsequent analysis. Indeed, it is prudent to start the b-chain at this MDL solution as then the necessary burn-in time can be greatly reduced.

A.7 Dimensionality discussion

Many vertex feature often take only one of several discrete values within a set. For example, with the primary school dataset [17], a pupil's school-class can take one of 10 values from {"1A", "1B", "2A", "2B" ... "5A", "5B"}.

It is not immediately obvious how to encode this for the feature-to-block classifier. Although this is technically only a single dimension, we choose to expand the data into a set of 10 binary feature flags all taking values in the set $\mathcal{X} = \{0, 1\}$. Although only one flag will be set at a time it is the simplest method for representing discrete-valued data. The reason we choose $\mathcal{X} = \{0, 1\}$ and not $\mathcal{X} = \{-1, 1\}$ is that the former allows for simpler analysis of the resulting weight values; when a feature d is switched off then $x_d = 0$ given $\mathcal{X} = \{0, 1\}$, and the feature has no impact on the softmax output. This would not be the case for $\mathcal{X} = \{-1, 1\}$.

Furthermore, it is more intuitive to only model positive relationships. In other words, we prefer to say that this block is comprised of the vertices with feature d turned on rather than this block contains the vertices with feature d switched off. To summarise the approach explicitly, say we have a feature encoding a pupil's gender: {"male", "female"}, we represent

this as 2 binary feature flags "male": {0, 1} and "female": {0, 1}. This approach may seem inefficient but it is vital to ensure we can interpret the resulting parameter distributions.

Nevertheless, this dimensionality expansion means that we should not include a bias term in our softmax as then the MAP solution is less peaked and our θ-samples will have higher variance. To illustrate this point we need only look at the form of the softmax activation function with bias vector β, operating on a vector x which only has one non-zero entry $x_d = 1$. Component j of the output is given by:

$$\phi_j(x) = \frac{\exp(w_j^T x + \beta_j)}{\sum_{k\in[B]} \exp(w_k^T x + \beta_k)} = \frac{\exp(w_{jd} + \beta_j)}{\sum_{k\in[B]} \exp(w_{kd} + \beta_k)}. \tag{40}$$

The term $w_{kd} + \beta_k$ is effectively the new weight for component k as these two terms will always appear together. The bias term β_k is therefore an unnecessary extra degree of freedom. To illustrate this point, we set $\beta_k = \beta_0$ for all j. If this is the case then equation 40 can be simplified to:

$$\phi_j(x) = \frac{\exp(\beta_0) \cdot \exp(w_{jd})}{\sum_{k\in[B]} \exp(\beta_0) \cdot \exp(w_{kd})} = \frac{\exp(w_{jd})}{\sum_{k\in[B]} \exp(w_{kd})} \tag{41}$$

This expression is independent of β_0 and so it is free to vary. This is not the whole picture as we also have the prior which introduces a regularisation term which favours weights closer to 0. This ensures the MAP solution is unique. Nevertheless, for mutually exclusive binary feature flags, the bias term does not add to the expressiveness of the model and only serves to complicate analysis. We therefore remove the bias term from the softmax.

Even for data comprised of several feature-classes each expanded into flags (e.g. school class and gender) we prefer to discard the bias term. The bias term only serves to decrease the objective function by leveraging information about the size of each detected block and not feature information. We do not wish to be overly confident in our block predictions when such predictions are influenced by block size and not feature information. Indeed, the approach was initially developed with a bias term but we found its removal yielded more reliable and reproducible results.

A.8 Hypothesis test on feature weights

We are given samples $\{\theta^{(t)}\}_{t\in\mathcal{T}_\theta} \sim p(\theta|A, X)$ and wish to determine the statistical significance of the weights. We adopt matrix notation for simplicity by representing θ with the matrix $B \times D$ matrix of feature weights W. The question is to determine the significance of a particular feature d by examining the value of W_{id} for all the values $i \in [B]$. We know that

the posterior is proportional to the prior multiplied by the likelihood,

$$p(\theta|A, X) \propto p(\theta) \cdot p(A|\theta, X), \tag{42}$$

assuming that the feature matrix X is already given. The prior term can be evaluated but the likelihood is intractable. The closest we can get is through a Monte-Carlo integration over b,

$$\begin{aligned} p(A|\theta, X) &= \sum_{b\in[B]^N} p(A, b|\theta, X) \\ &= \sum_{b\in[B]^N} p(A|b, \theta, X) \cdot p(b|\theta, X) \\ &= \sum_{b\in[B]^N} p(A|b) \cdot p(b|\theta, X) \\ &\approx \sum_i p\left(A|b^{(i)}\right) \quad \text{with} \quad b^{(i)} \sim p(b|\theta, X). \end{aligned} \tag{43}$$

This could be implemented for a single value of θ but such a form cannot be used to characterise the overall form of the posterior. Nevertheless, the form in (43) highlights something interesting. The likelihood is peaked around areas of θ that generate a partition b that is highly likely in the SBM sense – high $p(A|b)$. This provides the motivation for using the Laplace approximation for modelling the posterior $p(\theta|A, X) \approx p(\theta; \hat{\mu}, \hat{\Sigma})$; we use a multivariate Gaussian to approximate the posterior. Indeed, the Laplace approximation is often used for modelling the posterior in logistic classification [15]. This approximation is not exact but it motivates the derivation of the dimensionality reduction technique we construct by analogy with hypothesis testing. Marginalising over the other elements, the posterior for each element of the weight matrix W can be approximated by,

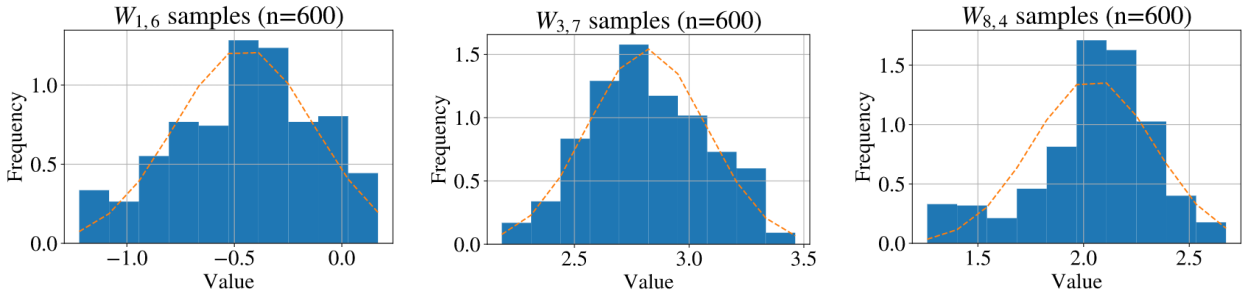
$$p(W_{ij}|A, X) \approx \mathcal{N}(W_{ij}|\hat{\mu}_{ij}, \hat{\sigma}^2_{ij}). \tag{44}$$

Where we have used the set of samples for W drawn according to the exact posterior, to calculate unbiased estimates for the mean and standard deviation:

$$\hat{\mu}_{ij} \coloneqq \frac{1}{|\mathcal{T}_\theta|} \sum_{t\in\mathcal{T}_\theta} W^{(t)}_{ij} \qquad \text{and} \qquad \hat{\sigma}^2_{ij} \coloneqq \frac{1}{|\mathcal{T}_\theta|} \sum_{t\in\mathcal{T}_\theta} \left(W^{(t)}_{ij} - \hat{\mu}_{ij}\right)^2. \tag{45}$$

This approximation is not exact but we can show it is accurate empirically. Indeed, if we run the primary school experiment with hyper-parameters given in Appendix C.2, we can then plot histograms of the collected W-samples and compare these to the Laplace approximation.

The results are given on Figure 9. Even though the Laplace approximation is not exact, it is remarkably reliable.



(a) Weight for block 1, class 3B (b) Weight for block 3, class 4A (c) Weight for block 8, class 2B

Figure 9: Histograms of sampled weights for the primary school experiment [17]. Dotted line is the applied Laplace approximation.

Using the approximation, we wish to construct a test to determine whether a particular weight value has $|W_{ij}| > c$ with high probability, where $c > 0$. More specifically, we wish to decide between:

$$\begin{aligned} H_0 &: |W_{ij}| \leq c \\ H_1 &: |W_{ij}| > c \end{aligned} \tag{46}$$

We can consider the probabilities $p(W_{ij} > c)$ and $p(W_{ij} < c)$ separately. Starting with $p(W_{ij} > c)$, we write this as,

$$\begin{aligned} p(W_{ij} > c) &= p\left(Z > \frac{c - \hat{\mu}_{ij}}{\hat{\sigma}_{ij}}\right) \\ &= p\left(-Z < \frac{\hat{\mu}_{ij} - c}{\hat{\sigma}_{ij}}\right) \\ &= \Phi\left(\frac{\hat{\mu}_{ij} - c}{\hat{\sigma}_{ij}}\right), \end{aligned}$$

where $Z \sim \mathcal{N}(0, 1)$ and $\Phi(\cdot)$ is the standard Gaussian cumulative density function. We introduce the variable k to control the degree of significance of the result. Enforcing k to be less than the argument of Φ gives us that:

$$\hat{\mu}_{ij} - k\hat{\sigma}_{ij} > c.$$

As $\Phi(\cdot)$ is monotonically increasing, this yields the following:

$$p(W_{ij} > c) > \Phi(k) \iff \hat{\mu}_{ij} - k\hat{\sigma}_{ij} > c. \tag{47}$$

A similar argument can be made to show that:

$$p(W_{ij} < -c) > \Phi(k) \iff \hat{\mu}_{ij} + k\hat{\sigma}_{ij} < -c. \tag{48}$$

Clearly equations 47 and 48 cannot both hold simultaneously as $c, \hat{\sigma}_{ij}, k > 0$. Nevertheless, a necessary and sufficient condition for one of them to hold is:

$$(\hat{\mu}_{ij} - k\hat{\sigma}_{ij}, \hat{\mu}_{ij} + k\hat{\sigma}_{ij}) \cap (-c, +c) = \emptyset \tag{49}$$

This is equivalent to saying that $\hat{\mu}_{ij}$ is at least k standard deviations away from the range $(-c, c)$. We choose to reject the null hypothesis H_0 if and only if equation 49 holds. We can say with probability greater than or equal to $\Phi(k)$ that either $W_{ij} < -c$ or $W_{ij} > c$ thus rejecting the null $H_0 : |W_{ij}| \leq c$.

Note that we choose to accept H_0 even in the case that the combined probabilities of $p(W_{ij} < -c) + p(W_{ij} > c) > \Phi(k)$, if equation 49 is not satisfied. This is because we wish to apply this for dimensionality reduction purposes. We only want to keep weights for which we have a confident prediction outside the range $(-c, c)$ not those that succeed by summing the two extremes of their distribution. Note that this case is very rare empirically but it nonetheless must be specified.

B Derivations

B.1 Derivation of $p(b|X)$

We determine the form of $p(b|X)$ by integrating out the parameters θ. From the definitions, we have:

$$\begin{aligned} p(b|X) &= \int p(b,\theta|X,\theta)d\theta = \int p(b|X,\theta)p(\theta|X)d\theta \\ &= \int p(b|X,\theta)p(\theta)d\theta = \int \prod_{i\in[N]} \phi_{b_i}(x_i;\theta)p(\theta)d\theta \\ &= \prod_{i\in[N]} \int \frac{\exp(w_{b_i}^T x_i)\prod_{j\in[B]}\mathcal{N}(w_j;0,\sigma_\theta^2 I)}{\sum_{k\in[B]}\exp(w_k^T x_i)} dw_{1:B}. \end{aligned}$$

The key observation here is that the value of the integral is independent of the value of $b_i \in [B]$ as the integrand has the same form regardless of b_i. This is because the prior is the same for each w_j. Therefore, the integral is only a function of x_i and σ_θ^2, which means that, as a function of b, $p(b|X) \propto 1$. As b takes values in $[B]^N$, we necessarily have:

$$p(b|X) = \frac{1}{\left|[B]^N\right|} = B^{-N}. \tag{50}$$

B.2 Derivation of $U(\theta)$

Recall from (15) in Section 4.2 that,

$$\pi_\theta(\theta) \propto p(\theta|X,b) \propto p(b|X,\theta)p(\theta) \propto \exp\left(-U(\theta)\right),$$

so that U can be expressed as,

$$U(\theta) = -\left(\log p(b|X,\theta) + \log p(\theta)\right) + \text{const.}$$

Writing, $y_{ij} \coloneqq \mathbb{1}\{b_i = j\}$ and $a_{ij} \coloneqq \phi_j(x_i;\theta)$, we have that,

$$\log p(b|X,\theta) = \sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\log a_{ij} \quad \text{and} \quad \log p(\theta) = -\frac{(D+1)(B)}{2}\log 2\pi - \frac{1}{2\sigma_\theta^2}\|\theta\|^2, \tag{51}$$

where $\|\theta\|^2 = \sum_i \theta_i^2 = \sum_{j\in[B]} \|w_j\|^2$ is the Euclidean norm of the vector of parameters θ. Therefore, discarding constant terms, we obtain:

$$U(\theta) = \left(\sum_{i\in[N]} \sum_{j\in[B]} y_{ij} \log \frac{1}{a_{ij}} \right) + \frac{1}{2\sigma_\theta^2} \|\theta\|^2. \tag{52}$$

This is exactly the representation (16), as claimed.

B.3 Derivation of $\nabla U(\theta)$

Here we show how the gradient $\nabla U(\theta)$ can be computed explicitly. Recall the expression for $U(\theta)$ in (16). Writing θ as $\theta = \left[w_1^T, w_2^T \dots w_B^T\right]^T$, in order to compute the gradient $\nabla U(\theta)$ we need to compute each of its components, $\partial U / \partial w_k$, $1 \le k \le B$. To that end, we first compute,

$$\begin{aligned} \frac{\partial a_{ij}}{\partial w_k} &= \frac{\tilde{x}_i \exp(w_j^T \tilde{x}_i)\delta_{jk} \cdot \sum_{r\in[B]} \exp(w_r^T \tilde{x}_i) - \exp(w_j^T \tilde{x}_i) \cdot \tilde{x}_i \exp(w_k^T \tilde{x}_i)}{\left(\sum_{r\in[B]} \exp(w_r^T \tilde{x}_i)\right)^2} \\ &= \tilde{x}_i \left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right), \end{aligned} \tag{53}$$

where $\delta_{jk} \coloneqq \mathbb{1}\{j = k\}$, and we also easily find,

$$\frac{\partial}{\partial w_k} \|\theta\|^2 = \frac{\partial}{\partial w_k} \left(\sum_{r\in[B]} \|w_r\|^2 \right) = 2w_k. \tag{54}$$

Using (53) and (54), we obtain,

$$\begin{aligned} \frac{\partial U}{\partial w_k} &= \sum_{i\in[N]} \sum_{j\in[B]} y_{ij} \left(-\frac{\tilde{x}_i}{a_{ij}} \left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right) \right) + \frac{w_k}{\sigma_\theta^2} \\ &= -\left(\sum_{i\in[N]} \tilde{x}_i \left(y_{ik} - a_{ik} \sum_{j\in[B]} y_{ij} \right) - \frac{w_k}{\sigma_\theta^2} \right) \\ &= -\left(\sum_{i\in[N]} \left\{ \tilde{x}_i (y_{ik} - a_{ik}) \right\} - \frac{w_k}{\sigma_\theta^2} \right). \end{aligned} \tag{55}$$

This can be computed efficiently through matrix operations. The only property of y_{ij} we have used in the derivation is the constraint $\sum_{j\in[B]} y_{ij} = 1$, for all i.

C Implementation details

C.1 Algorithms

Algorithm 1 Block membership sample generation

procedure SAMPLEBLOCKMEMBERSHIPS(A, T_b)
  $b^{(0)} \leftarrow \arg\min_b S(b|A)$ ▷ Implemented as greedy heuristic in *graph-tool* library
  for $t \in \{0, 1 \dots T_b - 1\}$ **do**
    $b' \leftarrow\sim q_b(b^{(t)}, b'|A)$
    $\log \alpha_b \leftarrow \log \alpha_b(b^{(t)}, b'|A)$
    $\eta \leftarrow\sim \text{Unif}(0, 1)$
    if $\log \eta < \log \alpha_b$ **then**
      $b^{(t+1)} \leftarrow b'$
    else
      $b^{(t+1)} \leftarrow b^{(t)}$
    end if
  end for
  return $\{b^{(t)}\}_{t=1}^{T_b}$
end procedure

Algorithm 2 FFBM parameter pseudo-marginal inference

procedure SAMPLEFEATUREWEIGHTS($X, \{b^{(t)}\}, \mathcal{T}_b, \sigma_\theta, s$)
  $\hat{Y}_{ij} \leftarrow \frac{1}{|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} \mathbb{1}\{b_i^{(t)} = j\} \quad \forall i, j$
  $\theta^{(0)} \leftarrow\sim \mathcal{N}(0, \sigma_\theta I)$

  for $t \in \{0, 1 \dots T_\theta - 1\}$ **do**
    $\xi \leftarrow\sim \mathcal{N}(0, I)$
    $h_t \leftarrow \frac{s}{N} \cdot 250(1000 + t)^{-0.8}$
    $g_t \leftarrow \nabla U(\theta^{(t)} | X, \hat{Y})$

    $\theta' \leftarrow \theta^{(t)} - h_t \cdot g_t + \sqrt{2h_t} \cdot \xi$
    $\log \alpha_\theta \leftarrow \log \alpha_\theta(\theta^{(t)}, \theta' | A, \hat{Y})$
    $\eta \leftarrow\sim \text{Unif}(0, 1)$
    if $\log \eta < \log \alpha_\theta$ **then**
      $\theta^{(t+1)} \leftarrow \theta'$
    else
      $\theta^{(t+1)} \leftarrow \theta^{(t)}$
    end if
  end for
  return $\{\theta^{(t)}\}_{t=1}^{T_\theta}$
end procedure

Algorithm 3 Dimensionality reduction

procedure ReduceDimension($\{W^{(t)}\}, \mathcal{T}_\theta, k, D'$)
  $(B, D) \leftarrow W^{(0)}$.shape
  $\hat{\mu}_{ij} \leftarrow \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} W_{ij}^{(t)} \quad \forall i \in [B], j \in [D]$
  $\hat{\sigma}_{ij} \leftarrow \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} \left(W_{ij}^{(t)} - \hat{\mu}_{ij} \right)^2 \quad \forall i \in [B], j \in [D]$

  for $d \in [D]$ **do**
    for $i \in [B]$ **do**
      $l_i \leftarrow \hat{\mu}_{id} - k \cdot \hat{\sigma}_{id}$
      $u_i \leftarrow \hat{\mu}_{id} + k \cdot \hat{\sigma}_{id}$
      if $l_i \leq 0$ and $u_i \geq 0$ **then**
        $l_i, u_i \leftarrow 0$
      end if
    end for
    $c_d \leftarrow \max_i \min(|l_i|, |u_i|)$
  end for

  indexArray $\leftarrow$ indexSort(c, descending=True)$[0 : D']$
  $d^* \leftarrow$ indexArray[-1]
  $\mathcal{D}' \leftarrow$ Set(indexArray)
  $c^* \leftarrow c_{d^*}$
  return $\mathcal{D}', c^*$
end procedure

C.2 Hyperparameter values

Table 3: Hyper-parameter values for each experiment

Dataset	B	f	σ_θ	T_b	κ_b	λ_b	T_θ	κ_θ	λ_θ	s	k	D'	T'_θ	κ'_θ	λ'_θ	s'
Polbooks	3	0.7	1	1,000	0.2	5	10,000	0.4	10	0.05	–	–	–	–	–	–
School	10	0.7	1	1,000	0.2	5	10,000	0.4	10	0.2	1	10	10,000	0.4	10	0.2
FB Egonet	10	0.7	1	1,000	0.2	5	10,000	0.4	10	0.017	1	10	10,000	0.4	10	0.5

C.3 Hardware specification

All data analysis and visualisation was implemented in Python. Full source code is available in the supplementary material. The scripts were run using a standard PC using the Windows Subsystem for Linux (WSL) environment. Specs are:

- **CPU**: Intel(R) Core(TM) i7-1065G7
- **RAM**: 8GB
- **GPU**: Intel(R) Iris(R) Plus Graphics

On this hardware each experiment iteration took the following amount of time to execute:

Table 4: Compute-time for each experiment

Dataset	b-chain	θ-chain	Reduced θ-chain	Overall compute time
Polbooks	∼1s	∼10s	–	∼11s
School	∼1s	∼7s	∼7s	∼15s
FB Egonet	∼2s	∼50s	∼8s	∼60s

D Risk assessment retrospective

As experimental work was entirely computer-based the risks encountered were not what one might typically associate with a more hands-on project: heavy machinery, dangerous chemicals etc. Most of the risks identified would be as a result of an improper working environment. These could be properly managed by ensuring the working environment felt comfortable. An external monitor went some way to alleviating the eye-strain of coding on a small screen for extended periods. Furthermore, an external mouse is far kinder on the wrist than a track-pad on laptops. There were no unexpected hazards encountered that were not identified at the start of the project.

E Covid-19 disruption

Thankfully this project was not directly disrupted by Covid-19 as it was always going to be heavily theoretical and computational. The only disruption to speak of was the shift of supervisor meetings to online. Though shared whiteboards were alien at first, we grew to learn how to use them to effectively communicate ideas.

I was ill with Covid-19 for 2 weeks in October but this did not majorly impact my ability to work. The Covid-19 contingency plan formulated last summer and documented on the Project Planning Form just involved switching supervisions to an online format.