The Feature First Block Model

Lawrence Tray[1] and Ioannis Kontoyiannis[2]

[1] Department of Engineering, University of Cambridge, lpt30@cantab.ac.uk
[2] Statistical Laboratory, DPMMS, University of Cambridge, ik355@cam.ac.uk

Abstract. Labelled networks are a very common and important class of data. A typical inference goal is to determine how the vertex labels (or *features*) affect the network's 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 between 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 facilitates the use of richer queries on labelled network structure. 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.
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.

Keywords: Labelled Networks · Bayesian Inference.

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) [16]. 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 **(author?)** [13]. To allow for degree-variability between members of the same block, we adopt the degree-corrected formulation (DC-SBM), defined in (1).

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

Definition 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 $B \times B$ 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* $i \in [N]$*:*

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

We use the following notation to indicate this distribution:

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

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 vertices, 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 for 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,

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

Note that ϕ_{b_i} has 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.

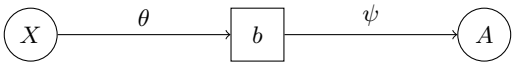


Fig. 1: The feature-first block model (FFBM)

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

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

Figure 1 illustrates the proposed model.

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 A.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. **(author?)** [13] 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. 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 through a Markov chain Monte-Carlo approach via a Metropolis-Hastings algorithm [5], as a two-level Markov chain.

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, **(author?)** [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.

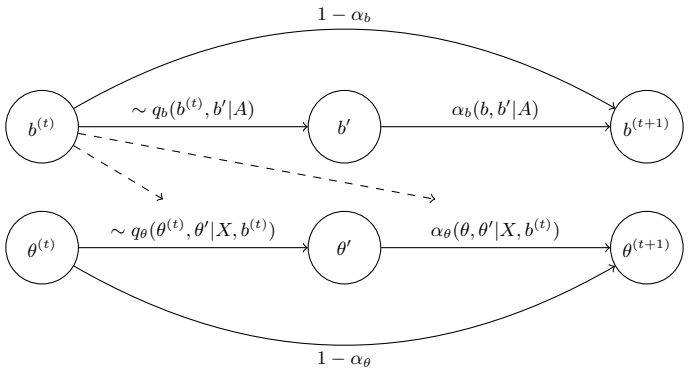


Fig. 2: Sampling sequence.

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.

4.1 Sampling block memberships

We adopt the Markov chain Monte-Carlo procedure of [10], 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; if we constructed an optimal code for the SBM ensemble, this would be the minimum amount of information required to store the graph and its parameters. The exact form of the proposal q_b is explored thoroughly in [10] and not repeated here. There is a widely used library for Python made available under LGPL called `graph-tool` [11], 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. We define $y_{ij} \coloneqq \mathbb{1}\{b_i = j\}$ and $a_{ij} \coloneqq \phi_j(x_i; \theta)$. Discarding constant terms, we can then 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 A.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 overfitting. 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) [14]. 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. 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 A.3). By noting that $\theta = \{w_k\}_{k=1}^{B}$, we write the derivative with respect to each w_k as,

$$\frac{\partial U}{\partial w_k} = -\left(\sum_{i\in[N]} \Big\{x_i(y_{ik} - a_{ik})\Big\} - \frac{w_k}{\sigma_\theta^2}\right). \tag{17}$$

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. 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.

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 then 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 $(\hat{\mu}_{ij} - k\hat{\sigma}_{ij}, \hat{\mu}_{ij} + k\hat{\sigma}_{ij})$ overlaps with $(-c, c)$ for all block indices i. If we were to use the Laplace approximation for the posterior $p(W_{ij}|A, X) \approx \mathcal{N}(W_{ij}; \hat{\mu}_{ij}, \hat{\sigma}^2_{ij})$, then this is analogous to a hypothesis test on the value of W_{ij} as in (23); then $\mathcal{D}'$ comprises all features j for which H_1 is accepted at least once for some $i \in [B]$.

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

The multiplier k in (21) determines the degree of significance of the result. However, as the Laplace approximation is not exact, we 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.

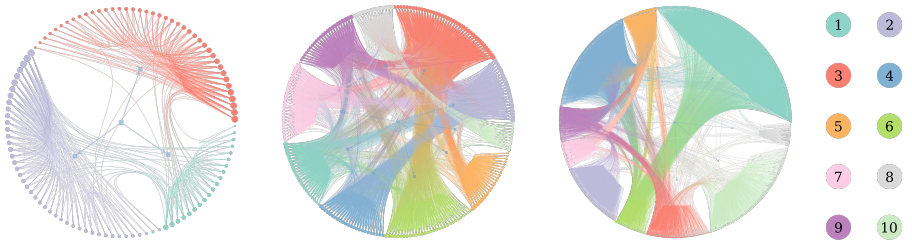
$$c^* = \arg\max\{c > 0 \ : \ |\mathcal{D}'| = D', k = k_0\}. \tag{24}$$

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** [15] ($N = 238, E = 5539, D = 13$) – 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. 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) Block Legend

Fig. 3: Networks laid out and coloured according to inferred block memberships $\hat{y}$ for a given experiment iteration. Visualisation performed using *graph-tool* [11].

Table 1: Experimental results averaged over $n = 10$ iterations (mean $\pm$ std. dev.).

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

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).

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 1 summarises the results for each experiment. 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 given in Table 1.

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. This requires defining the set of all vertices associated with block j, 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,

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

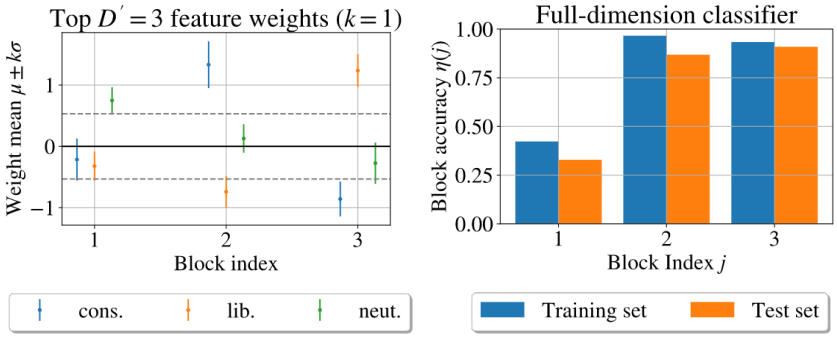
This is effectively testing whether the feature-to-block predictions and the graph-based predictions agree in their largest component. We call this metric, $\eta_\star(j)$, the *block-accuracy* for block j. It is clearly bounded $0 \leq \eta_\star(j) \leq 1$, with an accuracy of 1 meaning perfect agreement for the vertices in detected block 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. We discuss the results specific to each dataset in turn.

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).



(a) θ-samples. Dotted line is $\pm c^*$. (b) Per-block accuracy $\eta(j)$.

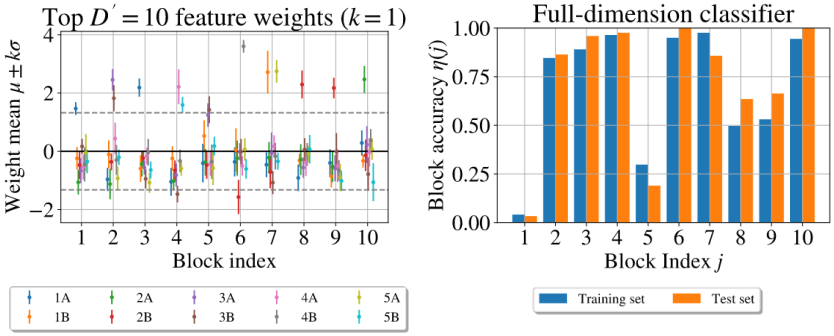
Fig. 4: Political books dataset.

From Figure 4a we see that all 3 blocks have a distinct political affiliation as their 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 1 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.

However, from Figure 4b we see that block 1 has low accuracy. This suggests that detected block 1 is not solely composed of "neutral" books but also contains some "liberal" and "conservative" authors. Indeed, by examining Figure 3a we see that block 1 is effectively a buffer between blocks 2 and 3; there are very few edges between blocks 2 and 3. It is therefore not surprising that some books from either side leak into block 1. Perhaps, the three distinct categories ("conservative", "liberal", "neutral") are too coarse a measure of political affiliation; it may be the case that the nominally "conservative" books found in block 1 are in fact more centre-right. In the absence of more granular labels, we cannot test this hypothesis. Nevertheless, political affiliation encoded as 3 distinct labels remains a fantastic predictor of 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.



(a) θ-samples. Dotted line is $\pm c^*$. (b) Per-block accuracy $\eta(j)$.

Fig. 5: Primary school dynamic contacts dataset.

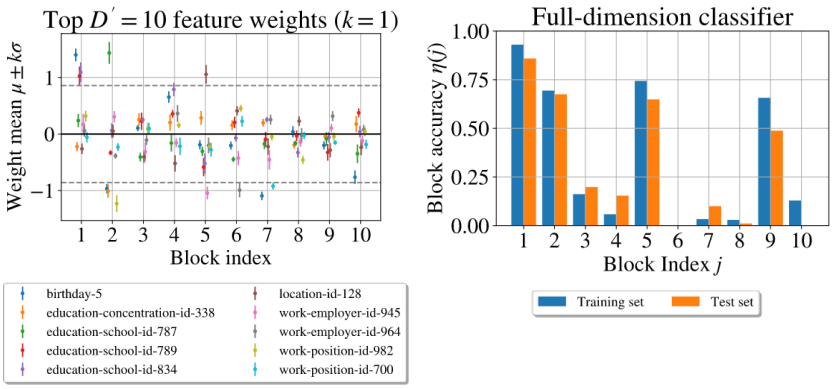
The vast majority of blocks are composed of a single class. However, some blocks have two comparably strong classes as their predictors. For example, blocks 2 and 5 both contain classes 3A and 3B as their 2 best predictors. This suggests that the social divide between classes is less pronounced for pupils in year 3. Conversely, some classes are found to extend over two detected blocks (class 2B spans blocks 8 and 9) but we do not have a feature which explains the division. The most surprising block is number 7, 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.

Figure 5b shows excellent accuracy for the majority of blocks. In fact the only blocks with low accuracy are those that have a school-class span two blocks such that we cannot reliably distinguish between the two. This is much more pronounced when we apply hard classification rather than the soft cross-entropy loss. Indeed block 1 has low accuracy because class 1A has a higher weight component for block 3 than it does for block 1. As we use hard classification to compute accuracy, vertices belonging to class 1A are predicted to belong to block 3. A similar effect is seen in block 5 which often loses out to block 2. The same can be said of blocks 8 and 9. However, in this case the weights for class 2B are

very similar – thus explaining the roughly evenly split accuracy for blocks 8 and 9.

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 $D' = 10$. This is further emphasised by the disparate accuracies in Figure 6b. Nevertheless, observe that the accuracy is only high for blocks that contain high magnitude weights. The only exception to this rule is block 9 which nonetheless may have high magnitude weights just below the cut-off c^*.



(a) θ-samples. Dotted line is $\pm c^*$. (b) Per-block accuracy $\eta(j)$.

Fig. 6: Facebook egonet dataset.

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 may 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. This issue could have been alleviated by choosing $\mathcal{X} = \{-1, +1\}$ such that having a feature turned off supports excluding a vertex from the block. However, Figure 6a demonstrates why we made the deliberate choice $\mathcal{X} = \{0, 1\}$. Because of the choice $\mathcal{X} = \{0, 1\}$, block 1 is allowed to contain 3 distinct feature categories rather than just one. We accept this trade-off: risking spurious high magnitude components (e.g. `birthday-5`) for the benefit of allowing a top-level block to span

multiple feature-groups. Nevertheless, one would have to apply a hierarchical approach to subdivide each top-level block into sub-blocks to test whether the constituent feature-groups determine structure at the lower-level.

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 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 determine 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 of the FFBM. 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.

A Derivations

A.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:

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

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 can only be a function of at most 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:

$$p(b|X) = \frac{1}{\left|[B]^N\right|} = B^{-N}. \tag{29}$$

A.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,

$$\log p(b|X,\theta) = \sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\log a_{ij} \quad \text{and} \quad \log p(\theta) = -\frac{DB}{2}\log 2\pi - \frac{1}{2\sigma_\theta^2}\|\theta\|^2, \tag{30}$$

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:

$$U(\theta) = \left(\sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\log\frac{1}{a_{ij}}\right) + \frac{1}{2\sigma_\theta^2}\|\theta\|^2. \tag{31}$$

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

A.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,

$$\begin{aligned}\frac{\partial a_{ij}}{\partial w_k} &= \frac{x_i \exp(w_j^T x_i)\delta_{jk} \cdot \sum_{r\in[B]} \exp(w_r^T x_i) - \exp(w_j^T x_i) \cdot x_i \exp(w_k^T x_i)}{\left(\sum_{r\in[B]} \exp(w_r^T x_i)\right)^2} \\ &= x_i \left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right), \end{aligned} \tag{32}$$

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

$$\frac{\partial}{\partial w_k}\|\theta\|^2 = \frac{\partial}{\partial w_k}\left(\sum_{r\in[B]} \|w_r\|^2\right) = 2w_k. \tag{33}$$

Using (32) and (33), we obtain,

$$\begin{aligned}\frac{\partial U}{\partial w_k} &= \sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\left(-\frac{x_i}{a_{ij}}\left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right)\right) + \frac{w_k}{\sigma_\theta^2} \\ &= -\left(\sum_{i\in[N]} x_i\left(y_{ik} - a_{ik}\sum_{j\in[B]} y_{ij}\right) - \frac{w_k}{\sigma_\theta^2}\right) \\ &= -\left(\sum_{i\in[N]} \Big\{x_i(y_{ik} - a_{ik})\Big\} - \frac{w_k}{\sigma_\theta^2}\right). \end{aligned} \tag{34}$$

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.

Bibliography

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

[2] Airoldi, E.M., Blei, D., Fienberg, S., Xing, E.: Mixed membership stochastic blockmodels. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems. vol. 21. Curran Associates, Inc. (2009), https://proceedings.neurips.cc/paper/2008/file/8613985ec49eb8f757ae6439e879bb2a-Paper.pdf

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

[4] Gaucher, S., Klopp, O., Robin, G.: Outliers detection in networks with missing links (2020)

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

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

[7] Mehta, N., Duke, L.C., Rai, P.: Stochastic blockmodels meet graph neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 4466–4474. PMLR (09–15 Jun 2019), http://proceedings.mlr.press/v97/mehta19a.html

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

[9] Pasternak, B., Ivask, I.: Four unpublished letters. Books Abroad **44**(2), 196–200 (1970), http://www.jstor.org/stable/40124305

[10] Peixoto, T.P.: Efficient monte carlo and greedy heuristic for the inference of stochastic block models. Physical Review E **89**(1) (Jan 2014). https://doi.org/10.1103/physreve.89.012804, http://dx.doi.org/10.1103/PhysRevE.89.012804

[11] Peixoto, T.P.: The graph-tool python library. figshare (2014). https://doi.org/10.6084/m9.figshare.1164194, http://figshare.com/articles/graph_tool/1164194

[12] Peixoto, T.P.: Hierarchical block structures and high-resolution model selection in large networks. Phys. Rev. X **4**, 011047 (Mar 2014).

https://doi.org/10.1103/PhysRevX.4.011047, https://link.aps.org/doi/10.1103/PhysRevX.4.011047

[13] Peixoto, T.P.: Nonparametric bayesian inference of the microcanonical stochastic block model. Physical Review E **95**(1) (Jan 2017). https://doi.org/10.1103/physreve.95.012317, http://dx.doi.org/10.1103/PhysRevE.95.012317

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

[15] Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.F., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., Vanhems, P.: High-resolution measurements of face-to-face contact patterns in a primary school. PLOS ONE **6**(8), 1–13 (08 2011). https://doi.org/10.1371/journal.pone.0023176, https://doi.org/10.1371/journal.pone.0023176

[16] Zhu, J., Song, J., Chen, B.: Max-margin nonparametric latent feature models for link prediction (2016)