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.
In this work, we introduce a new generative model, the feature-first block model (FFBM), to facilitate the use of richer queries on labelled network structure. We develop a Bayesian framework and devise a two-level Markov-Chain Monte-Carlo approach to efficiently sample the FFBM parameters. This allows us to infer if and how the observed vertex-features affect macro-structure.
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

Many real-world networks 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. In this work, we restrict our attention to vertex-labelled networks, referring to the labels as *features*. A typical goal is to determine whether a given feature impacts graphical structure. Answering this requires a random graph model; the standard is the stochastic block model (SBM) [5]. Numerous variants of the SBM have been proposed – such as the MMSBM [1] and OSBM [12]. However, these do not automatically include vertex features in the graph generation process.

To analyse a labelled network using one of the simple SBM variants, a typical procedure would be to partition the graph into blocks grouped by distinct values of the feature of interest. The associated model can then be used to test for evidence of heterogeneous connectivity between the feature-grouped blocks. Nevertheless, this approach can only consider disjoint feature sets and the feature-grouped blocks are often an unnatural partition of the graph.

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.

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). 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. Specifically, we will use the microcanonical variant of the SBM, proposed by Peixoto [9]. To allow for degree-variability between members of the same block, we adopt the degree-corrected formulation (DC-SBM), defined in (1).

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.*[3] *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. Let $k = (k_i)$ denote a vector of length N, 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. Specifically, if $A \sim \text{DC-SBM}_{\text{MC}}(b, e, k)$ then it must satisfy the following, for all $r, s \in [B]$ and all $i \in [N]$:

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

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 – illustrated in Figure 1).

Let N denote the number of vertices, B the number of blocks 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. We will later find it convenient to write θ as a $B \times D$ matrix of weights W. Specifically, the distribution of b given X and θ is,

$$p(b|X, \theta) = \prod_{i \in [N]} p(b_i|x_i, \theta) = \prod_{i \in [N]} \phi_{b_i}(x_i; \theta) = \prod_{i \in [N]} \frac{\exp\left(w_{b_i}^T x_i\right)}{\sum_{k \in [B]} \exp\left(w_k^T x_i\right)}. \tag{2}$$

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

Note that ϕ_{b_i} has the form of a softmax activation function. 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.

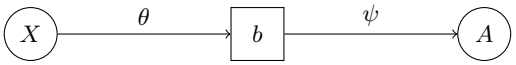


Fig. 1: The Feature-First Block Model (FFBM)

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

$$A \sim \text{DC-SBM}_{\text{MC}}(b, \psi_e, \psi_k). \tag{3}$$

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 such that each element of θ has an independent $\mathcal{N}(0, \sigma_\theta^2)$ prior, with hyperparameter σ_θ^2:

$$p(\theta) \sim \mathcal{N}\left(\theta; 0, \sigma_\theta^2 I\right). \tag{4}$$

This choice of prior gives a very simple form for the conditional distribution of the block membership vector b given X; it is a uniform distribution:

$$p(b|X) = \int p(b|X, \theta) p(\theta) d\theta = B^{-N}. \tag{5}$$

The proof is given in Appendix A.1. This an important simplification as evaluating $p(b|X)$ does not require an expensive integration over θ nor does it require the exact value of X. Peixoto [9] 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, $p(b, \psi|X) = p(b|X)p(\psi|b, X) = p(b|X)p(\psi|b)$, 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. Specifically, given a labelled network (A, X), we wish to infer if and how the observed features (X) impact graphical structure (A). Formally, this means characterising the posterior distribution: $p(\theta|A, X) \propto p(\theta) \cdot p(A|X, \theta)$.

Although the prior is easily computable, computing the likelihood would require a summation over all latent block-states, $p(A|X,\theta) = \sum_{b\in[B]^N} p(A|b)P(b|X,\theta)$, and is therefore intractable. This approach is made doubly intractable as we would also need to compute the normalising constant $p(A|X) = \int p(\theta)\cdot p(A|X,\theta)d\theta$. Clearly we cannot directly compute the form of the posterior, so instead we aim to draw samples from it,

$$\theta^{(t)} \sim p(\theta|A,X). \tag{6}$$

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

$$b^{(t)} \sim p\Big(b|A,X\Big) \quad \text{then} \quad \theta^{(t)} \sim p\left(\theta|X,b^{(t)}\right). \tag{7}$$

This can be implemented through a two-level Markov-Chain via the Metropolis-Hastings (MH) algorithm [3]. The splitting of the Markov-Chain into two levels allows us to side-step the summation over all latent $b \in [B]^N$ required to directly compute the likelihood, $p(A|X,\theta)$. The resulting $\theta^{(t)}$ samples are unbiased in that the expectation of their distribution is the true posterior:

$$\mathbb{E}_{b^{(t)}}\left[p\left(\theta|X,b^{(t)}\right)\right] = \sum_{b\in[B]^N} p(\theta|X,b)p(b|A,X) = p(\theta|A,X). \tag{8}$$

This is an example of a pseudo-marginal approach. Indeed, Andrieu and Roberts [2] show that (8) is sufficient to prove that, for large enough t, $\theta^{(t)} \sim p(\theta|A,X)$, as required.

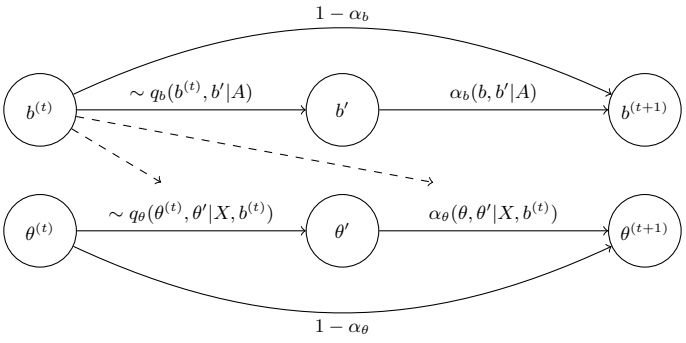


Fig. 2: θ-sample generation.

Figure 2 shows an overview of the proposed method, with q and α denoting the MH proposal distribution and acceptance probability respectively. Note the importance of the simplification in (5). As evaluating $p(b|X)$ does not depend on X, we do not need X to sample b. And on the other level, in order to obtain samples for θ we use only b but not A, as $(\theta \perp\!\!\!\perp A)|b$.

4.1 Sampling block memberships

To generate our b-samples, we adopt the Markov chain Monte-Carlo procedure of [7], which relies on writing the posterior in the following form:

$$p(b|A, X) \propto p(A|b, X) \cdot p(b|X) = \pi_b(b). \tag{9}$$

Now $\pi_b(\cdot)$ is the un-normalised target density. 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 needed to evaluate $p(A|b, X)$ reduces to just the single ψ^* term: $p(A|b, X) = \sum_\psi p(A, \psi|b, X) = p(A, \psi^*|b, X)$. We also define the microcanonical entropy of the configuration as,

$$S(b) \coloneqq -\log \pi_b(b) = -\Big(\log p(A|b, \psi^*) + \log p(\psi^*, b|X)\Big). \tag{10}$$

This can be thought of as the optimal "description length" of the graph. The exact form of the proposal q_b is explored thoroughly in [7] and not repeated here. We use the `graph-tool` [8] library for Python, which implements this algorithm. The only modification is in the 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,

$$\pi_\theta(\theta) \propto p(\theta|X, b) \propto p(b|X, \theta)p(\theta) \propto \exp\left(-U(\theta)\right), \tag{11}$$

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,

$$U(\theta) = \left(\sum_{i\in[N]}\sum_{j\in[B]} y_{ij} \log \frac{1}{a_{ij}}\right) + \frac{1}{2\sigma_\theta^2}\|\theta\|^2 = N \cdot \mathcal{L}(\theta) + \frac{1}{2\sigma_\theta^2}\|\theta\|^2. \tag{12}$$

See Appendix A.2 for the derivation. The function $U(\theta)$ is a typical objective function for neural network training. The first term $N \cdot \mathcal{L}(\theta)$ is introduced by the likelihood and represents the cross-entropy between the graph-predicted and feature-predicted block memberships. The second term, introduced by the prior, brings a form of regularisation, guarding against over-fitting. Our goal is to draw samples from the posterior $\pi_\theta(\cdot) \propto \exp(-U(\cdot))$. We choose to adopt the Metropolis-adjusted Langevin algorithm (MALA) [10], which uses ∇U to bias the proposal towards regions of higher density. Given the current sample θ, we generate a new sample θ' as,

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

where $\xi \sim \mathcal{N}(0, I)$ and h is a step-size parameter which may vary with the sample index. Without the injected noise term ξ, MALA is equivalent to gradient descent. We require ξ to fully explore the parameter space. The term ∇U has an easy to compute analytic form (derived in Appendix A.2).

4.3 Sampling sequence

So far, each $\theta^{(t)}$ update has used its corresponding $b^{(t)}$ sample. This means the evaluation of $U^{(t)}$ and $\nabla U^{(t)}$ has high variance, leading to longer burn-in. The only link between $b^{(t)}$ and $\theta^{(t)}$ is in the evaluation of $U^{(t)}$ and $\nabla U^{(t)}$ 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 $y_{ij}^{(t)}$:

$$\mathbb{E}\left[y_{ij}^{(t)}\right] = \mathbb{E}_{b^{(t)}}\left[\mathbb{1}\left(b_i^{(t)} = j\right)\right] = p(b_i = j|A, X). \tag{13}$$

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 $y_{ij}^{(t)}$ is then:

$$\hat{y}_{ij} \coloneqq \frac{1}{|\mathcal{T}_b|}\sum_{t\in\mathcal{T}_b} y_{ij}^{(t)} = \frac{1}{|\mathcal{T}_b|}\sum_{t\in\mathcal{T}_b} \mathbb{1}\{b_i^{(t)} = j\}. \tag{14}$$

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 $\hat{y}$ also allowing us to vary the lengths of each. This change shortens the burn-in of the θ-chain by reducing the variance of our proposal distribution.

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 to matrix notation $\theta = W$ we define:

$$\hat{\mu}_{ij} \coloneqq \frac{1}{|\mathcal{T}_\theta|}\sum_{t\in\mathcal{T}_\theta} W_{ij}^{(t)} \qquad \text{and} \qquad \hat{\sigma}_{ij}^2 \coloneqq \frac{1}{|\mathcal{T}_\theta|}\sum_{t\in\mathcal{T}_\theta}\left(W_{ij}^{(t)} - \hat{\mu}_{ij}\right)^2. \tag{15}$$

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 (16) and only keep features with indices $d \in \mathcal{D}'$, where $\mathcal{D}'$ is given in (17).

$$\mathcal{F}_i(j) \coloneqq (\hat{\mu}_{ij} - k\hat{\sigma}_{ij}, \hat{\mu}_{ij} + k\hat{\sigma}_{ij}) \cap (-c, +c), \tag{16}$$

$$\mathcal{D}' \coloneqq \{j \in [D] : \exists i \in [B] \text{ s.t. } \mathcal{F}_i(j) = \emptyset\} \tag{17}$$

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 $p(W_{ij}|A, X) \approx \mathcal{N}(W_{ij}; \hat{\mu}_{ij}, \hat{\sigma}_{ij}^2)$, then this would be analogous to a hypothesis test on the magnitude of W_{ij} compared to c with multiplier k in (16) determines the degree of significance of the result. However, as the Laplace approximation is not exact, we regard this as a useful heuristic and not an exact method. Conversely, if we want to fix the number of dimensions in our reduced feature set $|\mathcal{D}'| = D'$, the problem then becomes finding the largest value of c such that $|\mathcal{D}'| = D'$ given $k = k_0$:

$$c^* = \arg\max\{c > 0 \ : \ |\mathcal{D}'| = D', k = k_0\}. \tag{18}$$

5 Experiments

We apply the outlined methods to a variety of labelled networks. For reference, the inferred partitions for all of these are potted on Figure 3.

- **Political books** [6] ($N = 105, E = 441, D = 3$) – network of Amazon political book sales, published close to the 2004 presidential election. Two books are connected if they were frequently co-purchased. Vertex features encode the political affiliation of the author (liberal, conservative, or neutral).
- **Primary school dynamic contacts** [11] ($N = 238, E = 5539, D = 13$) – network of face-to-face contacts amongst students and teachers at a primary school in Lyon, France. 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** [4] ($N = 747, E = 30025, D = 480$) – an assortment of Facebook users' friends lists. Vertex features are fully anonymised and encode information about each user's education history, languages spoken, gender, home-town, birthday etc. We focus on the egonet with id 1912.

We first require metrics to assess model performance. We define the average description length per entity (nodes and edges) $\bar{S}_e$ as a suitable metric to gauge the SBM fit:

$$\bar{S}_e \coloneqq \frac{1}{(N+E)|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} S\left(b^{(t)}\right). \tag{19}$$

Next, to assess the performance of the feature-to-block predictor, we randomly partition the vertex set $[N]$ such that a constant fraction f forms our training set $\mathcal{G}_0$ and the remainder form our 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. As $|\mathcal{G}_0| \neq |\mathcal{G}_1|$ in general, we use the average cross-entropy loss over each set to gauge fit,

$$\bar{\mathcal{L}}_\star \coloneqq \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} \mathcal{L}_\star^{(t)}, \quad \text{where} \quad \mathcal{L}_\star^{(t)} \coloneqq \frac{1}{|\mathcal{G}_\star|} \sum_{i \in \mathcal{G}_\star} \sum_{j \in [B]} \hat{y}_{ij} \log \frac{1}{\phi_j\left(x_i; \theta^{(t)}\right)}, \tag{20}$$

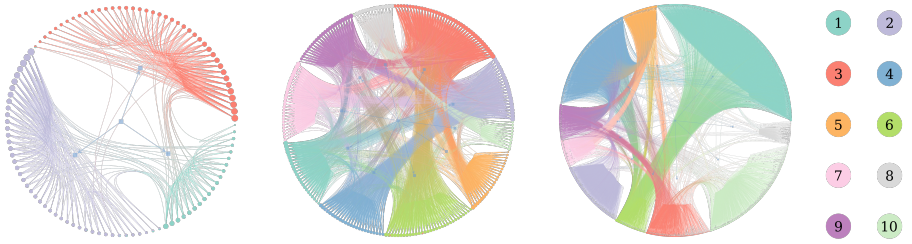
where $\star \in \{0, 1\}$ toggles between the training and test sets. Nevertheless, the cross-entropy loss is a coarse measure of fit. We wish to define a new measure of fit specific to each detected block. Let us define $\mathcal{B}_\star(j) \coloneqq \{i \in \mathcal{G}_\star : \hat{b}_i = j\}$ where $\hat{b}_i \coloneqq \arg\max_j \hat{y}_{ij}$. Now $\mathcal{B}_\star(j)$ is the set of vertices that have maximum a posteriori probability of belonging to block j. We now define the *block-accuracy* for block j as in 21. This effectively tests whether the feature-to-block and the graph-to-block predictions agree in their largest component.

$$\eta_\star(j) \coloneqq \frac{1}{|\mathcal{B}_\star(j)| \cdot |\mathcal{T}_\theta|} \sum_{i \in \mathcal{B}_\star(j)} \sum_{t \in \mathcal{T}_\theta} \mathbb{1}\left\{\hat{b}_i = \arg\max_j \phi_j\left(x_i; \theta^{(t)}\right)\right\}. \tag{21}$$

For the higher-dimensional datasets, we also apply the dimensionality reduction method of Section 4.4. We then retrain the feature-block predictor using

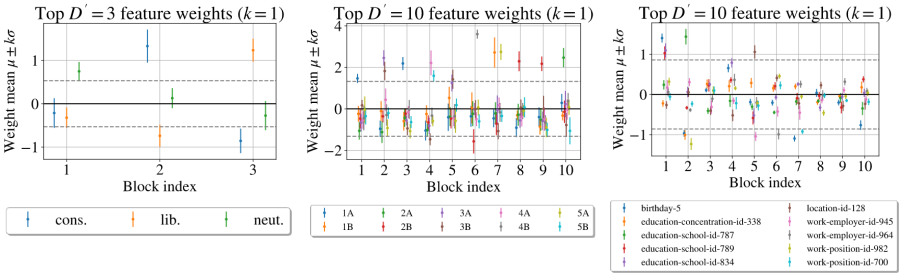
Table 1: Experimental results averaged over $n = 10$ iterations (mean $\pm$ std. dev.).

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



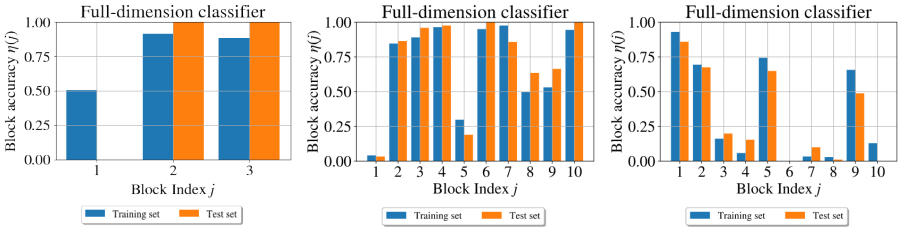
(a) Polbooks (b) School (c) Facebook egonet (d) Block Legend

Fig. 3: Networks laid out and coloured according to inferred block memberships $\hat{y}$ for a given experiment iteration. Visualisation performed using *graph-tool* [8].



(a) Political Books. (b) Primary School. (c) Facebook Egonet.

Fig. 4: θ-samples. Dotted line $\pm c^*$.



(a) Political Books. (b) Primary School. (c) Facebook Egonet.

Fig. 5: Per-block accuracy $\eta(j)$.

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.

Table 1 summarises the results for each experiment. We see 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).

Political books. 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 largest positive component. Furthermore, the training and test losses from Table 1 are very similar and both are low in magnitude. This is strong evidence that political affiliation is a very appropriate explanatory variable for the overall network structure. However, from Figure 5a 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. Examining Figure 3a, we see the majority of paths between blocks 2 and 3 go through block 1. Block 1 is in effect a bridge between the "conservative" and "liberal" blocks so it is unsurprising that some books from either side leak into block 1.

Primary school. We choose the number of communities $B = 10$, in line with the total number of school classes. Only the pupils' class memberships (1A-5B) survive the dimensionality-reduction process (Figur 4be); gender and teacher/student status have been discarded, meaning these are poor predictors of overall macro-structure. The vast majority of blocks are composed of a single class. However, some blocks have two comparably strong classes as their predictors (e.g. blocks 2 and 5). 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. 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 more pronounced when we apply hard classification rather than the soft cross-entropy loss. Perhapsthere are unobserved features which explain this divide.

Facebook egonet. We choose $B = 10$ and $D' = 10$ for this experiment. The selected features (Figure 4c) 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 some of the detected blocks; several blocks in Figure 4c do not have high-magnitude components for $D' = 10$. This is further emphasised by the disparate accuracies in Figure 5c.

6 Conclusion

The Feature-First Block Model (FFBM) introduced in this paper is a new generative model for labelled networks well-suited for describing how features affect 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 easy to find vertex features that are in some way correlated with the graphical structure; 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 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. 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.

A Appendix

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:

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

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:

$$p(b|X) = \frac{1}{\left|[B]^N\right|} = B^{-N}. \tag{22}$$

A.2 Derivation of $U(\theta)$ and $\nabla U(\theta)$

Recall from (11) 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,

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

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,

$$U(\theta) = \left(\sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\log\frac{1}{a_{ij}}\right) + \frac{1}{2\sigma_\theta^2}\|\theta\|^2. \tag{23}$$

As $\theta = \{w_k\}_{k=1}^B$, 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,

$$\begin{aligned}\frac{\partial a_{ij}}{\partial w_k} &= \frac{x_i \exp(w_j^T x_i)\delta_{jk}\cdot\sum_{r\in[B]}\exp(w_r^T x_i) - \exp(w_j^T x_i)\cdot x_i\exp(w_k^T x_i)}{\left(\sum_{r\in[B]}\exp(w_r^T x_i)\right)^2} \\ &= x_i\left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right), \end{aligned} \tag{24}$$

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

$$\frac{\partial}{\partial w_k}\|\theta\|^2 = \frac{\partial}{\partial w_k}\left(\sum_{r\in[B]}\|w_r\|^2\right) = 2w_k. \tag{25}$$

Recall the expression for $U(\theta)$ in (12). Using (24) and (25), we obtain,

$$\begin{aligned}\frac{\partial U}{\partial w_k} &= \sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\left(-\frac{x_i}{a_{ij}}\left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right)\right) + \frac{w_k}{\sigma_\theta^2} \\ &= -\left(\sum_{i\in[N]}\Big\{x_i(y_{ik} - a_{ik})\Big\} - \frac{w_k}{\sigma_\theta^2}\right). \end{aligned} \tag{26}$$

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.

A.3 Implementation details

All data analysis and visualisation was implemented in Python. Full source code is available at: `https://github.com/LozzaTray/Jormungandr-code`. Table 2 contains the hyper-parameters for each experiment run. The set of retained samples are generated as, $\mathcal{T}_\star = \{T_\star\kappa_\star + i\lambda_\star : 0 \le i \le \lfloor T_\star(1-\kappa_\star)/\lambda_\star \rfloor\}$, with κ_star controlling burn-in and λ_star controlling thinning.

Table 2: Hyper-parameter values for each experiment

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

Bibliography

[1] 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)

[2] Andrieu, C., Roberts, G.O.: The pseudo-marginal approach for efficient monte carlo computations. The Annals of Statistics **37**(2), 697 – 725 (2009)

[3] Hastings, W.K.: Monte carlo sampling methods using markov chains and their applications. Biometrika **57**(1), 97–109 (1970)

[4] 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)

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

[6] Pasternak, B., Ivask, I.: Four unpublished letters. Books Abroad **44**(2), 196–200 (1970)

[7] Peixoto, T.P.: Efficient monte carlo and greedy heuristic for the inference of stochastic block models. Physical Review E **89**(1) (Jan 2014)

[8] Peixoto, T.P.: The graph-tool python library. figshare (2014)

[9] Peixoto, T.P.: Nonparametric bayesian inference of the microcanonical stochastic block model. Physical Review E **95**(1) (Jan 2017)

[10] Roberts, G.O., Tweedie, R.L.: Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli **2**(4), 341 – 363 (1996)

[11] 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)

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