LATENT CHANNEL NETWORKS

CLIFFORD ANDERSON-BERGMAN

1. Abstract

Latent Euclidean embedding models a given network by representing each node in a Euclidean space, where the probability of two nodes sharing an edge is a function of the distances between the nodes. This implies that for two nodes to share an edge with high probability, they must be relatively close in *all* dimensions. This constraint may be overly restrictive for describing modern networks, in which having similarities in *at least* one area may be sufficient for having a high edge probability. We introduce a new model, which we call Latent Channel Networks, which allows for such features of a network. We present an EM algorithm for fitting the model, for which the computational complexity is linear in the number of edges and number of channels and apply the algorithm to both synthetic and classic network datasets.

2. Introduction

In this work, we define a graph $G = (N, E)$, where N is a set of nodes and E is an adjacency matrix, such that $E_{ij} = 1$ if nodes i and j share an edge and $E_{ij} = 0$ otherwise. At this time, we discuss undirected graphs, implying $E_{ij} = E_{ji}$ and ignore self loops, implying $E_{ii} = 0$. The degree of a node is defined as the number of edges attached to it. One classic example of this include social networks, in which nodes represent individuals and two individuals are considered to share an edge if they are listed as friends. Another common example is co-authorship graphs, in which nodes represent researchers and they are considered to share an edge if they have coauthored a paper together.

In the analysis of graph data, a common goal is to describe a network in a reduced order space, thereby providing insight of an underlying graph structure to the analyst. One of the simplest structures is the stochastic block model [11]. In this model, each node belongs to an unobserved block, where nodes have a fixed probability of having an edge with nodes within their block (p_{in}) and another fixed probability of having an edge with nodes outside their block (p_{out}). Typically, $p_{in} >> p_{out}$ so nodes are much more likely to share an edge with nodes within the same block, and each block can be considered a cluster. Recent work covers efficient estimation of the parameters of stochastic block models [1], [6], statistical characteristics of the estimators [14], [5] and model selection [25] and hierarchical stochastic block models [20].

One disadvantage to stochastic block models is they imply that within each block, the expected degree of a node is constant, with a variance implied by a binomial distribution. This fails to capture a very commonly observed phenomenon in social networks, namely that often a small number of nodes express an extremely high degree relative to most other nodes. In order to capture this, many other models have been proposed, such as the degree-corrected stochastic block model [13], in which edge probabilities is based on block membership *and* a given node's degree.

Another limitation of the stochastic block mode is that it is *hard clustering* approach. That is to say, each node deterministically belongs to a single block, and only one block. Several alternatives have been considered that allow for *soft clustering*. This includes the mixture stochastic block model [2], where each node belongs to a each block with a given probability. Another approach to tackle this problem by maximizing the modularity score [18], but with community membership described as a probability vector rather than a categorical variable [8], [9], [12]. In addition to modularity, other metrics such as overlapping correlation coefficient [4] may be used. We note that with the exception of [2], these methods are poised as purely optimization based clustering, rather a probability based model.

An alternative approach that ultimately motivated this work is that of a latent embedding. In a Euclidean embedding model [10], each node is represented in a latent Euclidean space, with edge probabilities being inversely proportional to distance. Because edge probabilities are directly modeled, one can naturally allow edge probabilities to be a function of both latent distance *and* linear predictors associated with each node. This model naturally allows for both very high and very low degree nodes; these are simply nodes whose intercept are exceptionally high or low. Traditional MCMC approaches are used for inference in [10], although method to accelerate this include using variational Bayes [22] and stratified case-control sampling [21]. A similar approach is that of a random dot product graph [19], [27], in which nodes a represented in a latent space and edge probabilties between two nodes are given by the dot product of their latent positionings. Estimates of the latent positions can be estimated via eigendecompositions of the adjacency matrix [23]. Clusters are not explicitly modeled in latent space embedding, but clustering may be performed on the lower-dimensional latent embedding.

The work we present is largely inspired by latent space embedding. One major disadvantage of an Euclidean embedding is that in order for two nodes to have a high edge probability, they must be close in *all* dimensions. However, in modern social networks it seems reasonable that being similar in *at least one* social dimension may be sufficient for high edge probability. To capture this dynamic, we present a Latent Channel Model, in which two nodes will share an observed edge in the graph if they are connected through at least one unobserved latent channel. The probability of two nodes connecting through a given channel is the product of each node's frequency of use of the given channel. If one considers channels to represent communities, our model can be viewed as something similar to a soft-clustering model. Under this interpretation, an important distinction between our model and other soft-clustering approaches that we are aware of is that we do not constrain community membership to sum to one. This very naturally models networks that contain a mix of high degree nodes (nodes that use multiple channels with high frequency) and low degree nodes (nodes that have low frequencies associated with all channels).

In section 3, we mathematically describe our model and present various ways to interpret meaningful parameters from the model. In section 4, we present both a simple and more complicated but more computationally efficient algorithm to compute the maximum likelihood estimate of the model parameters. In section 5, we apply the model to two stochastic block model networks and several classic real networks.

3. Latent Channel Model

3.1. **Model Parameterization.** Let use define an undirected graph G with nodes n_1,...,n_{N_n} and edges $e_{ij} = 1$ if n_i and n_j are connected and 0 otherwise. Define N_n to be the number of nodes and N_e to be the number of edges of the graph. We augment this observed graph with a latent set of channels C_1,...,C_K, which provide intermediate connections between nodes. In particular, we introduce latent edges $\tilde{e}_{ikj}$, which is equal to 1 if node n_i shares a latent edge to channel h_k toward node n_j. Our model dictates that a pair of nodes share an observed edge on the graph if they are both fully connected through one or more latent channel. More formally,

$$e_{ij} = \begin{cases} 1 & \text{if there exists } k \text{ such that } \tilde{e}_{ikj} = \tilde{e}_{jki} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

This is illustrated on figures 1 and 2. For simplicity, we define

$$c_{ijk} = \mathbb{I}(\tilde{e}_{ikj} = \tilde{e}_{jki} = 1). \tag{1}$$

In other words, c_{ijk} is an indicator that nodes i and j are connected through channel k.

We do not observe the $\tilde{e}_{ikj}$'s directly. However, our model dictates that for all j, $\tilde{e}_{ikj}$ are independently distributed Bernoulli distributions with probability p_{ik}. Thus, the marginal probability[1] that n_i will share a edge to n_j through channel h_k is $p_{ik}p_{jk}$. To compute the probability that nodes n_i and n_j share an edge, we compute

$$\begin{aligned} P(e_{ij} = 1) =& 1 - P(e_{ij} = 0) \\ & 1 - \prod_{k=1}^{K} (1 - P(\tilde{e}_{ikj} = 1 \cap \tilde{e}_{jki} = 1)) \\ & 1 - \prod_{k=1}^{K} (1 - p_{ik}p_{jk}) \end{aligned} \tag{2}$$

In other words, the probability nodes n_i and n_j share an edge is one minus the probability they do not share an edge. The probability they do not share an edge is the product of the probabilities they do not share an edge through any of the K channels.

As such, the log-likelihood of a latent channel graph can be written as

[1] Marginal probability meaning ignoring where n_i and n_j actually share an edge

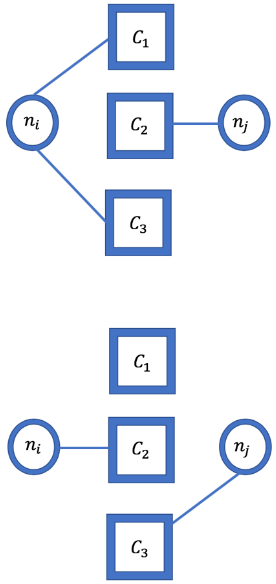

FIGURE 1. Nodes n_i and n_j do not share an edge as they are not connected through any channel.

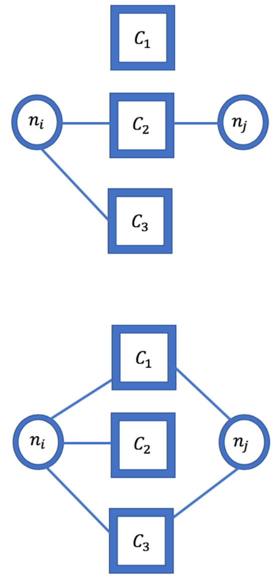

FIGURE 2. Nodes n_i and n_j share an edge as they are connected through at least one channel.

$$L(G|p) = \sum_{i=2}^{n} \sum_{j=1}^{i} e_{ij} \log \left(1 - \prod_{i=1}^{K} (1 - p_{ik}p_{jk})\right) + (1 - e_{ij}) \log \left(\prod_{i=1}^{K} (1 - p_{ik}p_{jk})\right) \tag{3}$$

3.2. **Interpretation of Model.** If one considers each channel to represent a latent community, then p_{ik} informally represents the strength of node i's attachment to community k. However, this parameter alone can be fairly hard to interpret, as it is unclear how large p_{ik}'s should be to be considered a strong connection. To help interpretation of the model, we present a few particularly interesting derived values.

We first consider parameter

$$\theta_{ijk} = P(c_{ijk} = 1 | e_{ij} = 1) = \frac{p_{ik}p_{jk}}{1 - \sum_{k=1}^{K} (1 - p_{ik}p_{jk})}. \tag{4}$$

The value θ_{ijk} represents the probability that nodes i and j are connected through channel k, given that the graph contains an edge between i and j. This is especially interesting in the case that channel k appears to have a meaningful interpretation, such as attachment strength parameter p_{ik} being correlated with meta-data on the nodes. For example, if attachment strength to channel k is strongly associated with nodes that have the occupation statistician, and θ_{ijk} is high, this suggests that given that nodes i and j

share a connection, they have a high probability of having an edge through the statistical community. It is worth noting that

$$\sum_{k=1}^{K} \theta_{ijk} \geq 1 \tag{5}$$

and typically with strict inequality. This is because two nodes that share an edge must share *at least* one edge through a latent channel, but may share many. For example, if channel k represents the statistical community and k' represents associations through a given research institution, statisticians at the same institution are likely to be connected through both channels k and k'.

Next, we consider

$$S_k = \sum_{i=1}^{N_n} p_{ik} \tag{6}$$

where we refer to S_k as the *size* of the channel. One way to interpret this parameter is that if a new node i' were to be fully connected to channel k, e.g. $p_{i'k} = 1$, it would be expected to have S_k connections through channel k. More generally, the expected number of connections for a new node would be $p_{i'k} S_k$.

Another particularly useful parameter is

$$C_{ik} = \mathbb{E}\left[\textstyle\sum_{j \neq i} c_{ijk} | G\right] = \sum_{j \neq i} e_{ij} \theta_{ijk}. \tag{7}$$

Formula 7 represents the expected number of connections node i has through channel k, conditional on the edges observed in the graph. While p_{ik} tells us the strength of attachment node i has to channel k, it is not sufficient to determine how many connections node i has through channel k. For example, a strong attachment to a small channel may result in fewer edges than a weak attachment to a large channel. As such, this statistic can provide insight in how many connections a node has through a given community, which is a function of both that individual's strength of attachment to the community *and* the size of the community.

Similar to equation 5, we note that

$$\mathbb{E}\left[\textstyle\sum_{j \neq i} c_{ijk} | G\right] \geq \sum_{j \neq i} e_{ij} \tag{8}$$

or that for subject i, the expected sum of connections through *all* channels is typically greater than the sum of all observed edges in the graph associated with that node. Again, this is because a single edge can be the result of connections through multiple channels.

We do suggest caution in over interpreting such parameters based on fitted data. As is the case for many probabilistic network models, we currently propose estimating the parameters via maximum likelihood estimation. Given the high dimensional parameter space, standard asymptotic normality results should not be considered a reliable method for estimating uncertainty. As such, we suggest using these methods for exploratory data analysis rather than making strong inference statements about a given network. Alternatively, Bayesian methods could be used to determine uncertainty. However, to do so, one must first address the unidentifiability issue that arises due to label switching of the channels.

4. Algorithm

We differ to maximum likelihood estimation to estimate the values of p_{ik}. In general, the problem is non-identifiable, as one can transpose the indices of the channels and arrive at an identical log-likelihood. Similarly, the problem is highly non-concave. As such, we will use an EM algorithm [7] to fit the parameters of the model.

4.1. **Fundamental EM Algorithm.** Note that if we observed the values of $\tilde{e}_{ikj}$, the log-likelihood would be greatly simplified to

$$L(G, \tilde{e}|p) = \sum_{i=1}^{N_n} \sum_{j \neq i}^{N_n} \sum_{k=1}^{K} \tilde{e}_{ikj} \log(p_{ik}) + (1 - \tilde{e}_{ikj}) \log(1 - p_{ik}) \tag{9}$$

which has closed form solution $\hat{p}_{ik} = \sum_{j \neq i}^{N_n} \tilde{e}_{ikj}/(N_n - 1)$, providing our M-step in the EM algorithm. For the E-step, we recognize that

$$P(\tilde{e}_{ikj} = 1 | e_{ij} = 1) = p_{ik}p_{jk} + p_{ik}(1 - p_{jk}) \left(1 - \prod_{k' \neq k} (1 - p_{ik'}p_{jk'})\right) \tag{10}$$

$$P(\tilde{e}_{ikj} = 1 | e_{ij} = 0) = p_{ik} - p_{ik}p_{jk} \tag{11}$$

We take an ECM algorithm [17] approach, where each p_{ik} is updated individually rather than all at once. This approach is particular advantageous when using caching with efficient updates. For clarity, we first present a simple, yet computationally inefficient, implementation in algorithm 1. Noting that computing $P(\tilde{e}_{ikj}|e_{ij} = 1)$ requires $O(K)$ operations and $P(\tilde{e}_{ikj}|e_{ij} = 0)$ requires $O(1)$ operations, this implementation of the algorithm requires $O(N_e K^2 + (N_n^2 - N_e)K)$ computations per iteration.

4.2. **Efficient Caching.** While the algorithm described in algorithm 1 is straightfoward, many of the computations in this algorithm are redundant and the order of complexity of this algorithm can be reduced by caching and updating various statistics.

For ease of notation, we define E_i to be the set of nodes that share an edge with node i and E_i^c to be the set of nodes that lack an edge with node i. We explicitly store $E_1, ..., E_{N_n}$ in an edge list, but do not explicitly store E_i^c. Note that we define i to be in neither E_i nor E_i^c.

Algorithm 1: Simple ECM Algorithm

Result: Fixed point estimate of $N \times K$ matrix p
Adjacency Matrix e; K;
N = nrow(e);
p = RandomUniform(min = 0, max = 1, nrow = N, ncol = K);
maxIters = 1,000; iter = 0; tol = 10^{-4}; maxDiff = tol + 1;
while *iter < maxIters & tol > maxDiff* **do**
  pOld = p;
  iter++;
  for *i in 1:N* **do**
    for *k in 1:K* **do**
      for *j in 1:N* **do**

$$\tilde{e}_{ijk} = \begin{cases} 0 & \text{if } i = j \\ P(\tilde{e}_{ikj} | e[i,j] = 1) & \text{else if } e[i,j] = 1 \\ P(\tilde{e}_{ikj} | e[i,j] = 0) & \text{otherwise} \end{cases}$$

      end
      p[i,k] = $\frac{\sum_{j=1}^{N} \tilde{e}_{ikj}}{N-1}$;
    end
  end
  maxDiff = max(| p - pOld |) ;
end
return (p)

We first note that the EM steps can be combined in the form

$$p_{ik}^{new} = \frac{\sum_{j \in E_i^c} P(\tilde{e}_{ikj} | e_{ij} = 0) + \sum_{j \in E_i} P(\tilde{e}_{ikj} | e_{ij} = 1)}{N - 1}. \tag{12}$$

We note that the first term of the numerator can be rearranged as

$$\begin{aligned} \sum_{j \in E_i^c} P(\tilde{e}_{ikj} | e_{ij} = 0) &= \sum_{j \in E_i^c} p_{ik} - p_{ik} p_{jk} \\ &= N p_{ik} (1 - \bar{p}_{.k}) - p_{ik} \left((1 - p_{ik}) + \sum_{j \in E_i} (1 - p_{jk}) \right) \end{aligned} \tag{13}$$

where $\bar{p}_{.k}$ represents the column mean of the p matrix.

Assuming $|E_i^c| > |E_i|$, this reduces the computation required to compute the first term from $O(|E_i^c|)$ to $O(|E_i|)$ as long as $\bar{p}_{.k}$ is cached. Note because the ECM algorithm only updates one entry of p at a time, each update only requires $O(1)$ operations to update the cached $\bar{p}_{\cdot k}$ at the end of each update.

Next, if we define

$$\pi_{ij} \equiv P(e_{ij} = 1) = 1 - \prod_{k=1}^{K}(1 - p_{ik}p_{jk}) \tag{14}$$

we can write

$$\begin{aligned}\sum_{j \in E_i} P(\tilde{e}_{ikj}|e_{ij} = 1) &= \frac{p_{ik}p_{jk} + p_{ik}(1-p_{jk})\left(1 - \prod_{k' \neq k}(1 - p_{ik'}p_{jk'})\right)}{\pi_{ij}} \\ &= \frac{p_{ik}p_{jk} + p_{ik}(1-p_{jk})\left(1 - \frac{1-\pi_{ij}}{1-p_{ik}p_{jk}}\right)}{\pi_{ij}}\end{aligned} \tag{15}$$

If the values π_{ij} are cached, this reduces the computations required for the second term of equation 12 from $O(K|E_i|)$ to $O(|E_i|)$. Note that if a single entry of p is updated, we can update π_{ij} in $O(1)$ time by computing

$$\pi_{ij}^{new} = 1 - \frac{(1-\pi_{ij})(1-p_{ik}^{new}p_{jk})}{1-p_{ik}^{old}p_{jk}}. \tag{16}$$

One technical note is that because we are considering an undirected graph, $\pi_{ij} = \pi_{ji}$ by definition. This implies that if we update p_{ik}, we must update both cache edge probabilities π_{ij} and π_{ji} unless they are explicitly saved and accessed as a single value. If π_{ij} is stored as a sparse matrix, this can be somewhat challenging to do in $O(1)$ time. We addressed this issue by storing the value of π_{ij} as a probability list P, where $P[i][j*]$ is the edge probability between node i and node i's j^{th} edge. We also created a mapping in advance that links $P[i][j*]$ to its corresponding transpose value, so that π_{ij} and π_{ji} can be updated in $O(1)$ time.

Finally, it should be noted that if $p_{ik} = 0$, then the EM algorithm will leave p_{ik} unchanged. As such, we can gain additional speedup by skipping the update for p_{ik} if $p_{ik} < \epsilon_p$ for a preset tolerance level ϵ_p.

We present pseudo code for the cached ECM algorithm in algorithm 2. The initial computational complexity of each of this algorithm is $O(K(N_n + N_e))$, although later steps of the algorithm can be significantly less by skipping updates where $p_{ik} < \epsilon_p$.

5. Applications

We demonstrate usage of the model on both synthetic and real datasets.

5.1. Stochastic Block Model.

First, we demonstrate usage of the model on a stochastic block model (SBM). In this case, we simulate an SBM with ten blocks, each with 100 nodes. For node pairs in the same block, the edge probability was set to 0.25. For node pairs in separate blocks, the edge probability was set to 0.025. Thus, on average, each node had 24.75 edges with nodes in the same block and 22.5 edges with nodes from other blocks.

Algorithm 2: Cached ECM Algorithm

```
Result: Fixed point estimate of N × K matrix p
Edge list E s.t. E[i][j] ≡ j^th index of node sharing j^th edge with node i;
ReverseMapping R, such that E[j][R[i][j]] = E[i][j];
p = RandomUniform(min = 0, max = 1, nrow = N, ncol = K);
pBar = ColumnMean(p);
initialize edge probability list edgeP, where edgeP[i][j] = P(e_{iE[i][j]} = 1);
maxIters = 10,000; iter = 0;
tol = 10^-3; pTol = 10^-10; maxDiff = tol + 1;
while iter < maxIters & tol > maxDiff do
    pOld = p;
    iter++;
    for i in 1:N do
        # Extract node indices and edge probabilities for nodes attached to node i
        theseEdges = E[i];
        theseEdgePs = edgeP[i];
        for k in 1:K do
            pik = p[i,k];
            if pik < pTol then
                skip;
            end
            # Compute contributions from nodes with and without edges to node i
            edgeContribution = 0.0;
            noEdgeContribution = N * pik * (1 - pBar[k]) - pik *(1 - pik);
            for j in theseEdges FIX THIS; SHOULD BE j in 1:leng... do
                pjk = p[j,k];
                noEdgeContribution -= pik * (1 - pjk);
                edgeContribution += pik*(pjk+(1-pjk)*(1- (1-theseEdgePs[j]) / (1-pik*pjk) )) /
                  theseEdgePs[j];
            end
            # Compute update
            pikNew = (edgeContribution + noEdgeContribution) / (N - 1);
            p[i,k] = pikNew;
            # Update column averages of p
            pikDiff = pikNew - pik;
            pBar[k] += pikDiff / N;
            for j in theseEdges do
                # Update edge probabilities, including transpose
                pjk = p[j,k];
                newEdgeP = 1 - ((1 - theseEdgePs[j]) * ( 1 - pikNew * pjk )) / (1 - pik * pjk);
                edgeP[i][j] = newEdgeP;
                edgeP[E[i][j]][R[i][j]] = newEdgeP;
            end
        end
    end
    maxDiff = max( | p - pOld | );
end
end
return p;
```

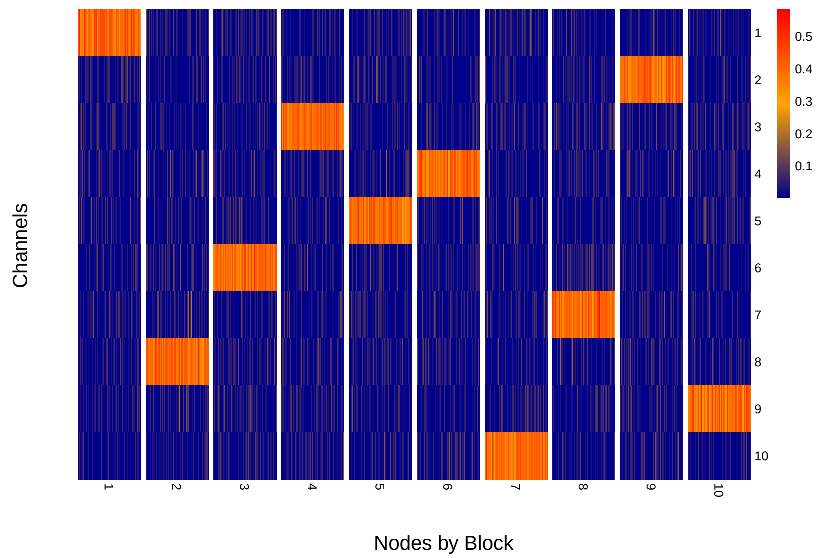

FIGURE 3. Latent channel model fit to a random stochastic block model. In each block, $p_{in} = 0.25$ and $p_{out} = 0.025$. Vertical blue lines are used to separate blocks.

We fit the latent channel model with ten channels and plotted a heat map on figure 3. We can see that the block structure is largely recovered, with some error.

One issue with standard SBMs is their sensitivity to high degree nodes. To emulate this, we augmented our original simulated SBM with one hundred new nodes that had an edge probability of 0.25 to *all* nodes of the graph. We refit the model and plotted on figure 4. We can see that the original structure remains largely intact, while the new high degree nodes are strongly attached to all of the latent channels.

5.2. **email-Eu-core Network.** An email network dataset was built between professors at a university, with edges existing if at least one email was sent between the two professors [26], [15]. The data was downloaded from the Stanford Large Network Dataset Collection [16]. This network contained 1005 nodes and 24,929 edges (after removing singular loops). In addition, the department of each professor was recorded. A total of forty two departments were listed, with department size ranging from 109 to 1. A histogram of department size can be found on figure 5.

We fit four latent channel models to this data with 5, 10, 20 and 40 channels. The results can be seen on 6. Because our model is a maximum likelihood estimator, AIC [3] can be used for model selection. Between our four models, the model with 10 channels has the lowest AIC.

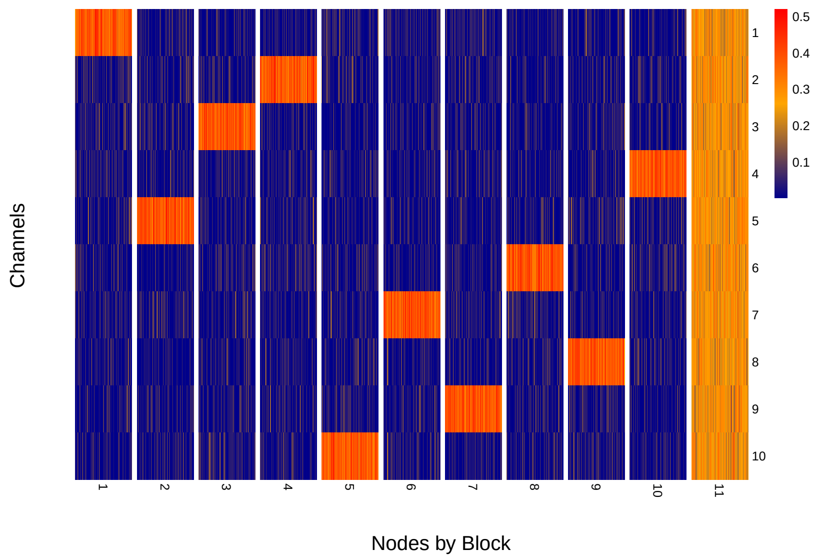

FIGURE 4. Latent channel model fit to a random stochastic block model, augmented with one hundred high degree nodes seen on the far right. Vertical blue lines are used to separate blocks. Note that the overall membership structure remains the same for all the original nodes, but the ten new nodes are strongly attached to every channel.

We examine the heatmap with 10 channels in more detail in 7. Within several of the departments, we see that several of the faculty are strongly attached to a similar channel. Clearly, not all 42 departments are fit to their own channel, but this is not very surprising given the large number of very small departments. Of each of the channels, the estimate channel sizes $\hat{S}_{ik}$ ranged from 39.6 to 62.0.

On figure 7, we have highlighted what we consider to be a particularly interesting department and channel. We see that within this department, several of the nodes are attached to *many* channels. In contrast, most other departments tend to only have a very few number of nodes strongly attached to more than one node. Similarly, there is one channel that is strongly attached to this department, which we have highlighted. We note that within other departments, there tend to be a small number of nodes attached to this channel. This suggests that this department communicates with other departments in very different ways. We hypothesize that this department could actually be an administrative group.

5.3. **Facebook100: UC Berkeley.** Next, we examine the UC Berkeley subset of the Facebook100 graph sets [24]. The UC-Berkeley graph contains 22,937 nodes with 852,445 edges. Several features are collected on each node. For this analysis, we consider gender, faculty/student status, major and graduation year.

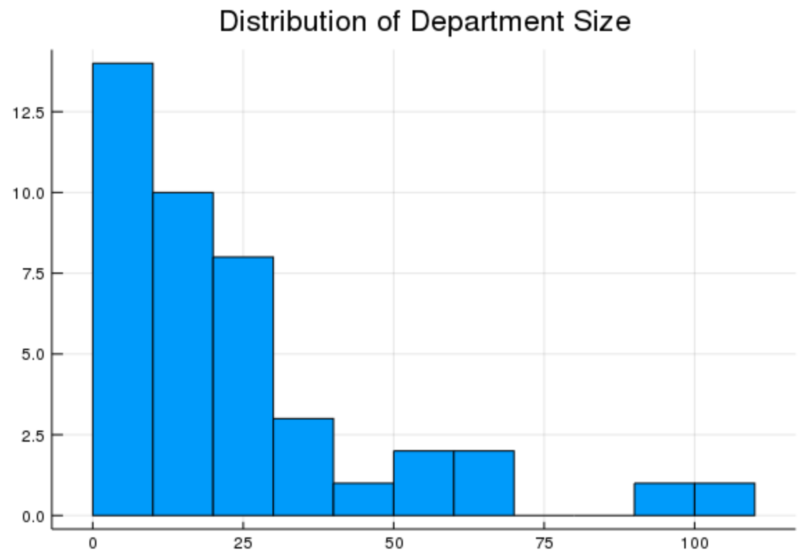

FIGURE 5. Distribution of department size for email-Eu-core network

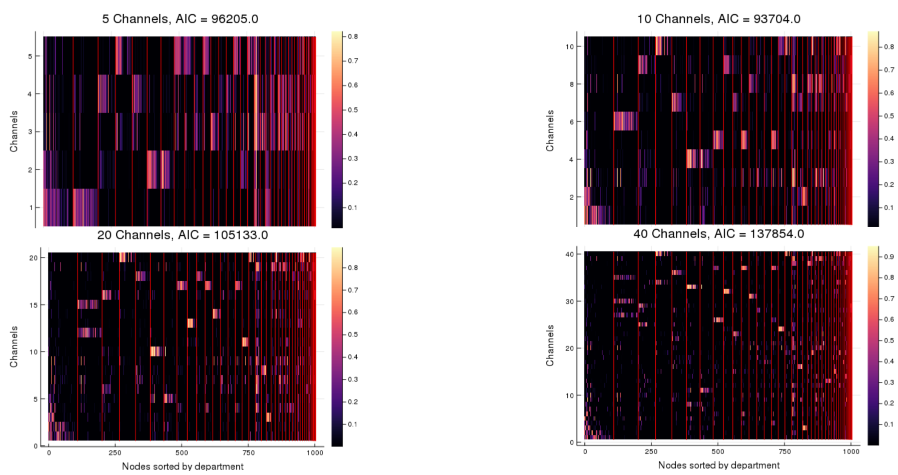

FIGURE 6. Email network with 5, 10, 20 and 40 channels. Vertical blue lines are used to separate departments.

Rather than selecting the number of channels via AIC

REFERENCES

[1] E. ABBE, A. S. BANDEIRA, AND G. HALL, *Exact recovery in the stochastic block model*, IEEE Transactions on Information Theory, 62 (2016), pp. 471–487.

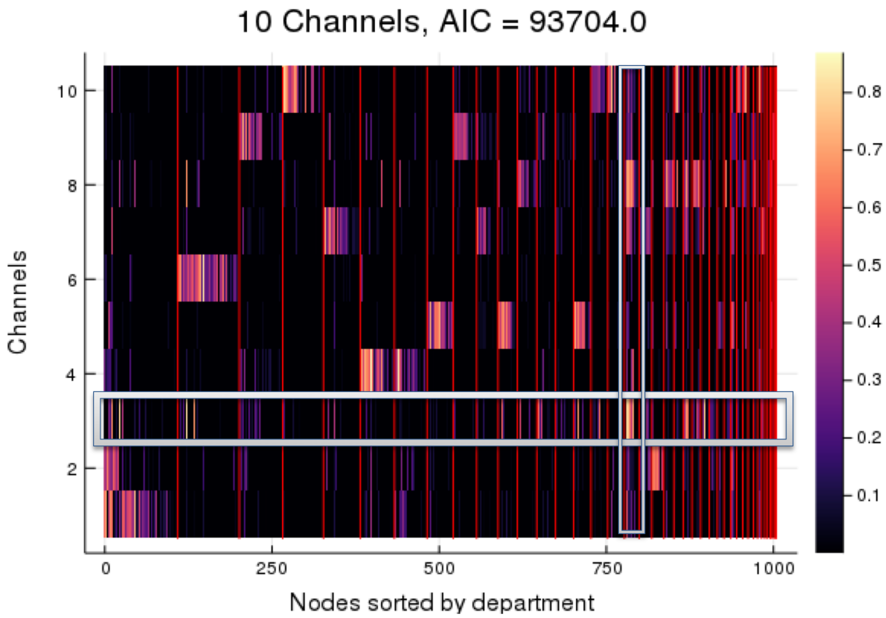

FIGURE 7. Email network with 10 channels.

[2] E. M. AIROLDI, D. M. BLEI, S. E. FIENBERG, AND E. P. XING, *Mixed membership stochastic blockmodels*, Journal of machine learning research, 9 (2008), pp. 1981–2014.

[3] H. AKAIKE, *Information theory and an extension of the maximum likelihood principle*, in Selected papers of hirotugu akaike, Springer, 1998, pp. 199–213.

[4] F. BONCHI, A. GIONIS, AND A. UKKONEN, *Overlapping correlation clustering*, Knowledge and information systems, 35 (2013), pp. 1–32.

[5] F. CALTAGIRONE, M. LELARGE, AND L. MIOLANE, *Recovering asymmetric communities in the stochastic block model*, IEEE Transactions on Network Science and Engineering, 5 (2018), pp. 237–246.

[6] P. CHIN, A. RAO, AND V. VU, *Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery*, in Conference on Learning Theory, 2015, pp. 391–423.

[7] A. P. DEMPSTER, N. M. LAIRD, AND D. B. RUBIN, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the Royal Statistical Society: Series B (Methodological), 39 (1977), pp. 1–22.

[8] E. GRIECHISCH AND A. PLUHÁR, *Community detection by using the extended modularity*, Acta cybernetica, 20 (2011), pp. 69–85.

[9] T. C. HAVENS, J. C. BEZDEK, C. LECKIE, K. RAMAMOHANARAO, AND M. PALANISWAMI, *A soft modularity function for detecting fuzzy communities in social networks*, IEEE Transactions on Fuzzy Systems, 21 (2013), pp. 1170–1175.

[10] P. D. HOFF, A. E. RAFTERY, AND M. S. HANDCOCK, *Latent space approaches to social network analysis*, Journal of the american Statistical association, 97 (2002), pp. 1090–1098.

[11] P. W. HOLLAND, K. B. LASKEY, AND S. LEINHARDT, *Stochastic blockmodels: First steps*, Social networks, 5 (1983), pp. 109–137.

[12] A. HOLLOCOU, T. BONALD, AND M. LELARGE, *Modularity-based sparse soft graph clustering*, in AISTATS 2019, 2019.

[13] B. KARRER AND M. E. NEWMAN, *Stochastic blockmodels and community structure in networks*, Physical review E, 83 (2011), p. 016107.

[14] J. Lei, A. Rinaldo, et al., *Consistency of spectral clustering in stochastic block models*, The Annals of Statistics, 43 (2015), pp. 215–237.

[15] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph evolution: Densification and shrinking diameters*, ACM Transactions on Knowledge Discovery from Data (TKDD), 1 (2007), p. 2.

[16] J. Leskovec and A. Krevl, *SNAP Datasets: Stanford large network dataset collection*. http://snap.stanford.edu/data, June 2014.

[17] X.-L. Meng and D. B. Rubin, *Maximum likelihood estimation via the ecm algorithm: A general framework*, Biometrika, 80 (1993), pp. 267–278.

[18] M. E. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Physical review E, 69 (2004), p. 026113.

[19] C. L. M. Nickel, *Random dot product graphs a model for social networks*, PhD thesis, Johns Hopkins University, 2008.

[20] T. P. Peixoto, *Hierarchical block structures and high-resolution model selection in large networks*, Physical Review X, 4 (2014), p. 011047.

[21] A. E. Raftery, X. Niu, P. D. Hoff, and K. Y. Yeung, *Fast inference for the latent space network model using a case-control approximate likelihood*, Journal of Computational and Graphical Statistics, 21 (2012), pp. 901–919.

[22] M. Salter-Townshend and T. B. Murphy, *Variational bayesian inference for the latent position cluster model for network data*, Computational Statistics & Data Analysis, 57 (2013), pp. 661–671.

[23] D. L. Sussman, M. Tang, and C. E. Priebe, *Consistent latent position estimation and vertex classification for random dot product graphs*, IEEE transactions on pattern analysis and machine intelligence, 36 (2014), pp. 48–57.

[24] A. L. Traud, P. J. Mucha, and M. A. Porter, *Social structure of facebook networks*, Physica A: Statistical Mechanics and its Applications, 391 (2012), pp. 4165–4180.

[25] Y. R. Wang, P. J. Bickel, et al., *Likelihood-based model selection for stochastic block models*, The Annals of Statistics, 45 (2017), pp. 500–528.

[26] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, *Local higher-order graph clustering*, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 555–564.

[27] S. J. Young and E. R. Scheinerman, *Random dot product graph models for social networks*, in International Workshop on Algorithms and Models for the Web-Graph, Springer, 2007, pp. 138–149.