Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

CHRISTIAN TOBIAS WILLENBOCKEL[1] AND CHRISTOF SCHÜTTE[1,2]

[1] *Department of Mathematics and Computer Science, Freie Universität Berlin, Germany*

[2] *Zuse Institute Berlin, Germany*

A Variational Bayesian Algorithm for Clustering of Large and Complex Networks

ZIB-Report 15–25 (April 2015)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

A Variational Bayesian Algorithm for Clustering of Large and Complex Networks

Christian Tobias Willenbockel * †[a] and Christof Schütte ‡[a,b]

[a] *Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 6, D-14195 Berlin, Germany*
[b] *Konrad–Zuse–Zentrum für Informationstechnik Berlin (ZIB), Takustraße 7, D-14195 Berlin, Germany*

Abstract

We propose the Blockloading algorithm for the clustering of large and complex graphs with tens of thousands of vertices according to a Stochastic Block Model (SBM). Blockloading is based on generalized Variational Bayesian EM (VBEM) schemes and works for weighted and unweighted graphs. Existing Variational (Bayesian) EM methods have to consider each possible number of clusters separately to determine the optimal number of clusters and are prone to converge to local optima making multiple restarts necessary. These factors impose a severe restriction on the size and complexity of graphs these methods can handle. In contrast, the Blockloading algorithm restricts restarts to subnetworks in a way that provides error correction of an existing cluster assignment. The number of clusters need not be specified in advance because Blockloading will return it as a result. We show that Blockloading outperforms all other variational methods regarding reliability of the results and computational efficiency.

Keywords: *Clustering*, *Variational Bayes EM*, *Model Selection*, *Stochastic Block Model*, *Networks*, *unsupervised classification*
MSC: Primary 62H30; Secondary 62H12

1 Introduction

The estimation of a Stochastic Block Model (SBM) [1, 2] for a given graph has become an established tool for model based clustering. The SBM is applied in different scientific areas like the Social Sciences or Biology. Often, only the graph of a network is given without further information. Estimation of the SBM consists of assigning the vertices of the graph to clusters and inferring the probabilities for the existence of an edge dependent of the cluster membership of the vertices. The results of the SBM can be easily interpreted and link prediction of edges follows easily.
In the last years, Variational methods like the Variational Expectation Maximization

*corresponding author
†Email: willenbo@mi.fu-berlin.de
‡Email: schuette@zib.de

(VEM) [3], [4] or Variational Bayesian Expectation Maximization (VBEM) algorithm [5], [6] were introduced for inferring the hidden cluster assignment of the vertices and to estimate the model parameters of the SBM for large graphs.
All these variational methods have in common, that they depend on starting values, i.e., their performance depends on the choice of the initial cluster partition matrix $\boldsymbol{Q}$. For finding the optimal number of clusters K^* using a variational method, the respective algorithm is initialized for different numbers of clusters K and initial cluster partition matrices $\boldsymbol{Q}$. For all K under consideration a variational bound, called Free Energy F of the SBM [7, 5] is calculated. Then the result with the optimal Free Energy and its corresponding cluster partition matrix $\boldsymbol{Q}^*$ and number of clusters K^* is chosen. In this algorithmic set-up, most of the computational time is used solely for determining the optimal number of clusters and only a fraction of the time is invested into calculating the optimal partition matrix $\boldsymbol{Q}^*$. Depending on the starting values, which are at least to a certain degree chosen randomly, this approach often only converges to a *local optimum* making multiple restarts for each number of clusters necessary [5, 6]. In particular, there is no algorithmic procedure for utilizing the information contained in already calculated partition matrices. These drawbacks limit the complexity and size of graphs variational methods can handle with meaningful results.
We propose the Blockloading algorithm for overcoming these problems and thus allowing application to graphs too large and complex for the existing VBEM algorithms. Blockloading is based on the VBEM set-up but starts from an initial cluster partition matrix $\boldsymbol{Q}$ with all vertices of the network in just one large cluster. Then Blockloading iteratively optimizes the VBEM functional while expanding the number of clusters until no further optimization is possible. This expansion is achieved by dividing one of the clusters of the respective current partition matrix, $\boldsymbol{Q}$, into two new clusters so that an initial guess of start values is needed only for the partition submatrices regarding the vertices in the respective cluster. This allows for a more careful choice of the start values. We will propose several algorithms for finding those start values and checking whether splitting an existing cluster really is the optimal choice in the framework of Blockloading. The general strategy is to reuse previously calculated results as starting values, because the Blockloading algorithm optimizes the vertices belonging to one active cluster with the other vertices and clusters kept fixed.
This article is organized in the following way:
In Section 2, we will review the Stochastic Block Model for unweighted and weighted graphs. Next, in Section 3, we discuss the Variational Bayesian Estimation of Poisson and Bernoulli-based SBMs for a given network, and we repeat the main steps of the VBEM algorithm. In Section 4 we present the exact Integrated Complete Likelihood criterion (ICL_{ex}) for comparing two SBM models for a given network for the Poisson SBM. Then, in Section 5, we propose the Blockloading algorithm and describe its different steps. This is complemented by Section 6, where several algorithms for finding the start values for the Blockloading algorithm are discussed. Finally, in Section 7, we demonstrate the performance of Blockloading in application to some test networks, including networks generated from an SBM where the model parameters and the cluster partition of vertices is known explicitly, but also an Earthquake Network [8] where the ground truth is not available. In all experiments made, Blockloading outperformed all other variational algorithms.

2 Stochastic Block Model

A graph $G = (V,E)$ consists of a set V of N vertices or vertices and a set of (directed) edges E connecting the vertices. The edges connecting the vertices are given by an adjacency matrix $\boldsymbol{A}$. If there is an edge from vertex i to vertex j it is $A_{ij} = 1$. If there is no edge from vertex i to vertex j, it is $A_{ij} = 0$. In the case of weighted graphs it holds $A_{ij} = w_{ij}$, $w_{ij} \in (0,1,2,\dots)$, if there is an edge from i to j. In this paper we well consider directed graphs unless otherwise stated.
The following Stochastic Block Model (SBM) was introduced in [1] as an algorithm for generating graphs. We assume that $\boldsymbol{A}$ was generated by the SBM. The SBM assigns the vertices V of the graph depending on their connection probability patterns to clusters. The SBM consists of K clusters. To each vertex i, the SBM assigns a unique cluster membership. A vertex belongs to cluster k with probability δ_k with $\sum_{k=1}^{K} \delta_k = 1$. The cluster membership is given by the random variable $\boldsymbol{Z}_i \in R^{1\times K}$, with $Z_{ik} = 1$ if i is an element of cluster k and $Z_{ik} = 0$ otherwise. $\boldsymbol{Z}$ is the $N \times K$ cluster indicator matrix with matrix rows $\boldsymbol{Z}_i$ for $i \in \{1,\dots,N\}$. An edge exists within each cluster k with the probability θ_{kk} and between cluster k and l with the probability θ_{kl}. So, the SBM is generated in the following way [5]:
(i) Roll a k – sided dice with $p(i \in k|Z_{ik} = 1) = \delta_k$ for side k for each vertex i, to determine the unique cluster membership of the vertex.
(iia) Flip a coin for each pair of vertices. With probability $\theta_{kl} = p(A_{ij}\,|\,Z_{ik}Z_{jl} = 1)$ there is an edge from vertex i to j with $i \in k$ and $j \in l$ and with probability $1-\theta_{kl}$ there is no edge.
Therefore we can sum up the joint probability by:

$$\begin{aligned} p(\boldsymbol{A},\boldsymbol{Z}|\boldsymbol{\delta},\boldsymbol{\theta},K) &= p(\boldsymbol{A}|\boldsymbol{Z},\boldsymbol{\theta},K)\,p(\boldsymbol{Z}|\boldsymbol{\delta},K) \\ &= \prod_{i\neq j}^{N}\prod_{k,l}^{K}\left(\theta_{kl}^{A_{ij}}(1-\theta_{kl})^{(1-A_{ij})}\right)^{Z_{ik}Z_{jl}} \prod_{i=1}^{N}\prod_{k=1}^{K}\delta_k^{Z_{ik}}. \end{aligned} \tag{1}$$

The results of the clustering are easily interpretable. The prediction of new edges with this model follows naturally from the estimated parameters. Variants of the SBM for directed and weighted graphs exists [4]. For example it is possible to replace the Bernoulli distribution in (1) with a Poisson distribution [4]:

$$f(A_{ij};\lambda_{kl}) = \frac{\lambda_{kl}^{A_{ij}}}{A_{ij}!}\exp\left(-\lambda_{kl}\right). \tag{2}$$

Then we can replace (iia) with:
(iib) Draw a realization from $f(\cdot;\lambda_{kl})$ for the edge A_{ij} from vertex i to vertex j, with $i \in k$ and $j \in l$. Then, the joint probability for directed graphs is:

$$p(\boldsymbol{A},\boldsymbol{Z}|\boldsymbol{\delta},\boldsymbol{\lambda},K) = \prod_{i\neq j}^{N}\prod_{k,l}^{K} f(A_{ij};\lambda_{kl})^{Z_{ik}Z_{jl}} \prod_{i=1}^{N}\prod_{k=1}^{K}\delta_k^{Z_{ik}}. \tag{3}$$

Using the Poisson distribution also works for unweigthed graphs. In the following, we call this SBM the Poisson SBM contrary to the Bernoulli SBM of [1].

3 Variational Bayesian Expectation Maximization

The Variational Bayesian Expectation Maximization algorithm (VBEM) is an important part of the Blockloading algorithm presented in Section 5. In Section 2, we re-

viewed how to generate a graph with the Bernoulli– or Poisson SBM. Now, we want to cluster a given graph according to the SBM. So, we have to solve the inverse problem of fitting the SBM to a given graph. The VBEM algorithm is used to calculate the unknown optimal number of clusters K, the parameters and the hidden cluster assignments of the vertices of the SBM. For the Bernoulli SBM outlined in Section 2, the VBEM algorithm for solving the SBM was presented in [6]. In the following, we present a VBEM algorithm for the Poisson SBM of Section 2. A Variational EM algorithm for the Poisson SBM was presented in [4].

The VBEM Algorithm is applied with a user specified number of clusters K. The input is the adjacency matrix $\boldsymbol{A}$. For convenience we write the parameters of the Bernoulli SBM as $\boldsymbol{\vartheta} = (\boldsymbol{\theta}, \boldsymbol{\delta})$ and likewise for the Poisson SBM $\boldsymbol{\vartheta} = (\boldsymbol{\lambda}, \boldsymbol{\delta})$.

For each vertex i, we want to calculate the latent cluster membership (latent variables) of the vertex, $\boldsymbol{Z}_i$ and the model parameters $\boldsymbol{\vartheta}$.

The optimal (true) number of clusters K^* is unknown. We can assume that $p(K)$ is a uniform distribution [5]. Thus, we can optimize $p(\boldsymbol{A}|K)$ instead of $p(K|\boldsymbol{A}) \propto p(\boldsymbol{A}|K)p(K)$ [5]. A variational lower bound of $-\ln p(\boldsymbol{A}|K)$, also called Free Energy F [5, 7], is obtained by using Jensen's inequality.

Variational distributions $q(\boldsymbol{Z}, \boldsymbol{\vartheta})$ over the parameters $\boldsymbol{\vartheta}$ and the latent variables $\boldsymbol{Z}$ are introduced. For obtaining a tractable algorithm, the mean field assumption $q(\boldsymbol{\vartheta}, \boldsymbol{Z}) = q(\boldsymbol{\vartheta})q(\boldsymbol{Z}) = q(\boldsymbol{\theta})q(\boldsymbol{\delta})\prod_{i=1}^{N} q(\boldsymbol{Z}_i)$ ([5, 6]) is assumed. This yields:

$$
\begin{aligned}
-\ln p(\boldsymbol{A}|K) &= -\ln \sum_{Z} \int p(\boldsymbol{A}, \boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{\delta}) p(\boldsymbol{\theta}) p(\boldsymbol{\delta}) \mathrm{d}\boldsymbol{\vartheta} && (4)\\
&= -\ln \sum_{Z} \int \frac{p(\boldsymbol{A}, \boldsymbol{Z}|\boldsymbol{\vartheta}) p(\boldsymbol{\vartheta})}{q(\boldsymbol{Z}) q(\boldsymbol{\vartheta})} q(\boldsymbol{Z}) q(\boldsymbol{\vartheta}) \mathrm{d}\boldsymbol{\vartheta} && (5)\\
&\leq -\sum_{Z} \int \ln \left(\frac{p(\boldsymbol{A}, \boldsymbol{Z}|\boldsymbol{\vartheta}) p(\boldsymbol{\vartheta})}{q(\boldsymbol{Z}, \boldsymbol{\vartheta})} \right) q(\boldsymbol{Z}, \boldsymbol{\vartheta}) \mathrm{d}\boldsymbol{\vartheta} && (6)\\
&\equiv F\left[q(\boldsymbol{Z}, \boldsymbol{\vartheta})\right]. && (7)
\end{aligned}
$$

We omitted K for better readability.

For the distribution $p(\boldsymbol{\theta})$ a Beta prior distribution $\mathrm{Beta}(\alpha_{kl}^0, \beta_{kl}^0)$ is specified, and for the distribution $p(\boldsymbol{\delta})$ a Dirichlet prior distribution $Dir(\boldsymbol{\delta}; \boldsymbol{\delta}^0)$. It was shown in [6], that $q(\boldsymbol{Z}_i) = \mathscr{M}(\boldsymbol{Z}_i; 1, \boldsymbol{Q}_i = \{Q_{i1}, \ldots, Q_{iK}\})$, where $\mathscr{M}$ is the Multinomial distribution, is the optimal approximation of $q(Z_i)$ for vertex i. The probability, that vertex i belongs to cluster k is given by Q_{ik}. It was also shown, that $q(\theta_{kl})$ has the functional form of a Beta distribution $Beta(\alpha_{kl}, \beta_{kl})$ and $q(\boldsymbol{\delta})$ the functional form of a Dirichlet distribution $Dir(\boldsymbol{\delta}; \boldsymbol{\delta}^0)$ [6]. We show in Proposition 3 in Appendix B.1 that the distributions $q(\lambda_{kl})$ have the functional form of a Gamma distribution $\mathrm{Gamma}(\lambda_{kl}; \alpha_{kl}, \beta_{kl})$.

To initialize the VBEM algorithm, we need start values for the latent variables Q_{ik}, $i \in \{1, \ldots, N\}$, $k \in \{1, \ldots, K\}$. The values Q_{ik} are entries of the $N \times K$ start partition matrix $\boldsymbol{Q}^{(start)}$.

The VBEM algorithm, introduced in [3] and [6] for the SBM in Section 2, consists now of the two following steps: In the Maximization Step (M-Step), the Free Energy F is optimized with respect to the distributions of the parameters $q(\boldsymbol{\vartheta})$, with the latent variables of the cluster memberships $q(\boldsymbol{Z})$ held fixed:

$$
\{q^{(t+1)}(\boldsymbol{\vartheta})\} = \underset{\{q(\boldsymbol{\vartheta})\}}{\arg\min}\, F[q^{(t)}(\boldsymbol{Z}), q^{(t)}(\boldsymbol{\vartheta})]. \tag{8}
$$

In the Expectation Step (E-step), F is optimized with respect to the latent variables $q(Z)$, with the distributions of the model parameters $q(\boldsymbol{\vartheta})$ held fixed:

$$\{q^{(t+1)}(\boldsymbol{Z})\} = \arg\min_{\{q(\boldsymbol{Z})\}} F[q^{(t)}(\boldsymbol{Z}), q^{(t+1)}(\boldsymbol{\vartheta})]. \tag{9}$$

The update equations for the Bernoulli SBM are discussed in Appendix A and the update equations of the Poisson SBM in Appendix B.1.
We calculate $F[q^{(t+1)}(\boldsymbol{Z}), q^{(t+1)}(\boldsymbol{\vartheta})]$ dependent on the latent cluster assignments $q^{(t+1)}(\boldsymbol{Z})$ and model parameters $q^{(t+1)}(\boldsymbol{\vartheta})$ which were returned by the M–and E–Step. The VBEM algorithm has converged, if

$$F[q^{(t)}(\boldsymbol{Z}), q^{(t)}(\boldsymbol{\vartheta})] - F[q^{(t+1)}(\boldsymbol{Z}), q^{(t+1)}(\boldsymbol{\vartheta})] < T \tag{10}$$

holds, where T is a threshold. Let $\boldsymbol{Q}^*$ be the cluster partition matrix and $\boldsymbol{\vartheta}^*$ the model parameters which were returned after convergence of the VBEM algorithm. Then, the value $F[\boldsymbol{Q}^*, \boldsymbol{\vartheta}^*] \equiv ILvb[\boldsymbol{Q}^*, \boldsymbol{\vartheta}^*]$ serves as a model selection criterion. This is the Integrated Likelihood Variational Bayes (ILvb) criterion introduced by [6] for the Bernoulli SBM. The ILvb for the Poisson SBM is

$$\begin{aligned} F[\boldsymbol{Q}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}] = &\sum_{k,l}^{K} \ln \left(\frac{\beta_{kl}^{\alpha_{kl}} \Gamma(\alpha_{kl}^0)}{\beta_{kl}^{0\,\alpha_{kl}^0} \Gamma(\alpha_{kl})} \right) + \sum_{i=1}^{N} \sum_{k=1}^{K} Q_{ik} \ln Q_{ik} \\ &+ \ln \left(\frac{\Gamma\left(\sum_{x=1}^{K} \delta_x\right) \prod_{x=1}^{K} \Gamma(\delta_x^0)}{\Gamma\left(\sum_{x=1}^{K} \delta_x^0\right) \prod_{x=1}^{K} \Gamma(\delta_x)} \right). \end{aligned} \tag{11}$$

It is proofed in Proposition 5. The ILvb allows us to compare different results of the VBEM algorithm. The lowest value of the ILvb corresponds to the optimal cluster assignment of the vertices of all calculations. The ILvb criterion of the Bernoulli SBM is repeated in Appendix A and the ILvb of the Poisson SBM Proposition 5.
To find the optimal number of clusters K^*, VBEM is initialized for different numbers of clusters K and different start partition matrices $\boldsymbol{Q}^{(start)}$. The optimal number of clusters is chosen as the calculation with the minimal value of the ILvb [6] (see also [5] for a closely related algorithm).

4 Exact Integrated Classification Likelihood Criterion

The algorithms for the initialization of the start partition matrix $\boldsymbol{Q}^{(start)}$ we will propose in Section 6 assign each vertex to exactly one cluster (hard clustering) contrary to the fuzzy clustering of the vertices by the VBEM algorithm in Section 3. We could also use the ILvb criterion for a hard clustering of the vertices. In this case, the entropy term, $\sum_{i=1}^{N} \sum_{x=1}^{K} Q_{ix} \ln Q_{ix}$, of eqn. (11) would be zero. The ILvb with the entropy term equal to zero is similar to the exact Integrated Classification Likelihood Criterion (ICL_{ex}), which was introduced in [9]. The derivation of the ICL_{ex} differs from the derivation of the ILvb. This can be seen when we proof the ICL_{ex} for the Poisson SBM in Appendix C or in [9] for the Bernoulli SBM. The ILvb criterion depends on the variational bound of the negative log likelihood, $-\ln\, p(A|K)$. So, the ILvb criterion is based on an approximation. Contrary, the ICL_{ex} is no approximation of the variational bound, but an analytical model selection criterion for the SBM [9]. It takes the cluster indicator matrix $\boldsymbol{Z}$ and the number of clusters K as the input. We present the ICL_{ex} in Proposition

1. We show in Proposition 4 how the model parameters $\boldsymbol{\vartheta}$ for a given cluster indicator matrix $\boldsymbol{Z}$ are calculated for the Poisson SBM.

Proposition 1. *Let the cluster indicator matrix $\boldsymbol{Z}$, the number of clusters K and the adjacency matrix A be given. Then the ICL_{ex} of the Poisson Stochastic Block Model of eqn. 3 is*

$$ICL_{ex}[\boldsymbol{Z},K] = \sum_{k,l}^{K} \ln\left(\frac{\beta_{kl}^{\alpha_{kl}}\Gamma(\alpha_{kl}^0)}{\beta_{kl}^{0\,\alpha_{kl}^0}\Gamma(\alpha_{kl})}\frac{1}{(A_{ij}!)^{\sum_{i\neq j}^{N} Z_{ik}Z_{jl}}}\right) + \ln\left(\frac{\Gamma\left(\sum_{k=1}^{K}\delta_k\right)\prod_{k=1}^{K}\Gamma(\delta_k^0)}{\Gamma\left(\sum_{k=1}^{K}\delta_k^0\right)\prod_{k=1}^{K}\Gamma(\delta_k)}\right) \tag{12}$$

where $\Gamma(\cdot)$ is the Gamma function. The parameters α_{kl}, β_{kl} and δ_k for all $k,l = 1,\ldots,K$ are calculated according to Proposition 3 given in Appendix B.1.

Every time we will use the ILvb criterion in Section 5 one can use the ICL_{ex} to compare hard clustered vertex partitions. This come at an increased computational cost. Another algorithm for finding the optimum of the ICL_{ex} for a given graph is the greedyICL algorithm proposed in [9]. It uses greedy heuristics introduced in [10] and [11], [9].

5 Blockloading Algorithm

Next, we propose the Blockloading algorithm, which is a cluster bisection algorithm. We start the optimization with all vertices of the graph in one cluster and calculate the Free Energy F (ILvb). This cluster is divided into two clusters in search for a lower Free Energy F . We continue to alternately divide and optimize chosen clusters by using a variant of the VBEM algorithm. This variant for the optimization of the vertices belonging to one cluster is called the BlockVB algorithm. The BlockVB algorithm for the Bernoulli SBM is given in appendix A and for the Poisson SBM in appendix B.1.

Initialization of Blockloading We calculate the Free Energy F for $K = 1$ clusters, i.e. all vertices are assigned to the same cluster. We use equations (28), (29), (30) together with eqn. (33) for the calculation of F of the Bernoulli SBM and Propositions 3, 4 and 5 for the Poisson SBM. The start partition matrix $\boldsymbol{Q}^{(start)}$ is an $N \times 1$–matrix with all entries equal to one. The result, the Free Energy F for $K = 1$, is stored as the reference $F^{(ref)}$.
Now, we check if a lower (better) Free Energy F than $F^{(ref)}$ can be obtained for a partition matrix with $K = 2$ clusters. A $N \times 2$–partition matrix $\boldsymbol{Q}^{(start)}$ is initialized for the VBEM algorithm of Section 3. There are several algorithms for initializing $\boldsymbol{Q}^{(start)}$. We explain these algorithms in Section 6. All these algorithms assign each vertices to exactly one cluster (hard clustering).
VBEM is run for $K = 2$ clusters and $\boldsymbol{Q}^{(start)}$ as input. The results are the partition matrix $\boldsymbol{Q}^{(trial)}$ and the Free Energy $F^{(trial)}$.
If $F^{(trial)} < F^{(ref)}$ holds, $\boldsymbol{Q}^{(trial)}$ is stored as the new reference partition matrix $\boldsymbol{Q}^{(ref)}$. Similarly, $F^{(ref)}$ is overwritten by $F^{(trial)}$ and the number of clusters is set to $K^{(ref)} = 2$. The model parameters calculated using Propositions 3 and 4 are stored in $\boldsymbol{\vartheta}_{ref}$.

After the Initialization Step, we check if there is a lower F for a partition of $K = 3$ clusters. We try to optimize and expand the existing reference partition $\boldsymbol{Q}^{(ref)}$ by dividing one of the existing clusters. There are several strategies to choosing a cluster we want to divide as explained in *The Active Cluster* below. To divide or optimize the assignment of the vertices in the active cluster, we need start values for the cluster assignment of the vertices. During tests we found that the quality of the results of the optimization of the active cluster depends strongly on the start values and that it is easier to find a (near) optimal partition of the vertices for a low number of clusters, K, like $K = 2$ than for a high number of clusters for example $K = 20$. We explain this finding with the help of the Largest Gap (LG) algorithm [12], which is presented in Section 6.2. A consequence of the derivation of the LG algorithm is, that for different cluster connection probabilities θ_{kl}, there are clusters which are easier to identify than other clusters. With the Optimal Gaps (OG) algorithm we will present in Section 6.2, these clusters are separated from the other clusters for a low number of clusters, starting with $K = 2$ clusters. Even for large and complex graphs, we are able to find near optimal start values for the vertex assignment for $K = 2$ clusters. This feature of the Blockloading algorithm also separates densely connected vertices from vertices which have comparatively few and irregular connections to other vertices. The choice of the active cluster allows us to proceed by optimizing the more densely connected parts of the network first. During tests we found that this optimization plan of the clusters yields a dramatic increase of the quality of the results especially for real world networks with highly varying connection profiles of the vertices. So, Blockloading allows the exclusion of vertices with sparse and irregular connections from the optimization process.
We use the results of the clustering for $K = 2$ clusters as start values for the following refinement and subdivision of the vertex assignment.
We sum up the main steps of the *Blockloading algorithm*:

Input: Adjacency matrix $\boldsymbol{A}$.
Result: Cluster partition matrix $\boldsymbol{Q}^{(ref)}$, number of clusters K_{ref} and parameters ϑ_{ref}.

(i) Blockloading Initialization

Main Loop:

(ii) Refinement Step

(iii) Expansion Step

(iv) Check for Convergence of all clusters

The Active Cluster We choose one of the existing clusters of $\boldsymbol{Q}^{(ref)}$ to split. We call this cluster the *active cluster*.
We choose the cluster with the highest number of vertices as the active cluster. The number of vertices in one cluster, n_k, are calculated according to:

$$n_k = \sum_{k=1}^{K} Q_{ik}^{(ref)}. \tag{13}$$

The active cluster a is therefore $a = \max_{k=1,\ldots,K} n_k$. In the following Refinement and Expansion Step it will become clear, that this choice minimizes the computational burden because we need fewer updates of the clusters before convergence is reached. For some

graphs with a high variance of vertex degrees, this choice is not optimal. This will be explained and alternatives will be introduced in Section 6.
The *Main Loop* of Blockloading begins with the Refinement Step of the active cluster a.

Hard Clustering As explained in Section 3, the partition matrix $Q^{(ref)}$ returns a probability for the cluster assignment of each vertex (fuzzy clustering). In the following paragraphs we introduce the concept of cluster dependent optimization of the existing partition matrix. To be able to optimize the vertices in one cluster, we first have to determine which vertices belong to which of the clusters. We transform the vertices i in cluster a of $\boldsymbol{Q}^{(ref)}$ into a hard clustering by transforming each matrix row $Q_i = (Q_{i1}^{(ref)}, \ldots, Q_{iK}^{(ref)})\ \forall i \in a$ according to

$$Q_{ij}^{(hard)} = \begin{cases} 1, & \text{if } Q_{ij} = \max\limits_{j \in \{1,\ldots,K\}} Q_{ij} \\ 0, & \text{else} \end{cases}. \tag{14}$$

Now, we know which vertex i belongs to the active cluster.

Refinement Step We determine the active cluster a and the set of indices I of all vertices with $i \in a$ (see eqn. 14). According to an algorithm of Section 6, we calculate cluster assignments for all vertices in I to the existing cluster of $\boldsymbol{Q}^{(ref)}$. All other vertices in $\boldsymbol{Q}^{(ref)}$ are kept fixed. This yields the start partition matrix $\boldsymbol{Q}^{(start)}$.
We run the BlockVB algorithm of Appendix A or B.1 with the matrix $\boldsymbol{Q}^{(start)}$, number of clusters K_{ref} and set of indices I. BlockVB is a variant of VBEM, which calculates only the cluster assignment for vertices $i \in I$ with the cluster assignment of all vertices $i \notin I$ held fixed.
BlockVB returns the fuzzy cluster assignment stored in the matrix $\boldsymbol{Q}^{(trial)}$, the Free Energy $F^{(trial)}$ and the parameters $\vartheta^{(trial)}$. If $F^{trial} < F^{ref}$ holds, the output $\boldsymbol{Q}^{(trial)}$, $F^{(trial)}$, and $\vartheta^{(trial)}$ of the BlockVB algorithm is stored as the new reference.
The change in the cluster assignment of vertices $i \in I$ can affect the cluster assignment of the vertices $i \notin I$ too, (see [3] for an example). Therefore, all clusters have to be updated until convergence is reached. Consequently we reset the number of converged clusters to zero.
If $F^{(trial)} \geq F^{(ref)}$ we keep the old reference values. We proceed to the Expansion Step.

One can also skip the Refinement Step for a considerable additional gain in computation speed. This comes at the cost of increased variance of the quality of the results for complicated networks.
The Refinement Step is an *error correction* feature of the Blockloading algorithm which is already used during the calculation process. When some algorithms for choosing the start partition matrix are used, the Refinement Step can also be skipped (see Section 6).

Expansion Step If there was a change of the Free Energy $F^{(ref)}$ in the Refinement Step, we determine the active cluster a and the set I. We try to find an F lower than the existing $F^{(ref)}$, by finding a cluster assignment of all vertices $i \in a$ to two new clusters. The reference matrix $\boldsymbol{Q}^{(ref)}$ is duplicated and renamed as $\boldsymbol{Q}^{(start)}$. An additional matrix column of zeros is added to the matrix $\boldsymbol{Q}^{(start)}$ making it an $N \times K^{(ref)} + 1$–matrix. We

apply one of the start value algorithms from Section 6 to the set of vertices I, to divide it into two clusters $k_1^{(I)}$ and $k_2^{(I)}$. The vertices $i \in k_1^{(I)}$ are kept in a. The vertices $i \in k_2^{(I)}$ are set to zero. Then, all matrix entries $\boldsymbol{Q}^{(start)}_{iK^{(ref)}+1}$ with $i \in k_2^{(I)}$ are set to one.
BlockVB is run with partition matrix $\boldsymbol{Q}^{(start)}$ and $K^{(ref)}+1$ as input. As explained in the Refinement Step, we get the output of $\boldsymbol{Q}^{(trial)}$, $\vartheta^{(trial)}$ and $F^{(trial)}$ from BlockVB. If $F^{(trial)} < F^{(ref)}$ holds, the reference values $F^{(ref)}$, $\boldsymbol{Q}^{(ref)}$ and $\boldsymbol{\vartheta}^{(ref)}$ are updated. Like in the Refinement Step, the vector of converged clusters is emptied because the update of the reference values can affect the convergence of the other clusters, too.
Otherwise, we store the active cluster a as *converged*.

Convergence Convergence is reached, if the optimization of all clusters in the Refinement Step and Expansion Step has converged. The Expansion or Refinement Step did not yield a lower (better) Free Energy F. The number of converged clusters is equal to $K^{(ref)}$.

The Blockloading algorithm may also be used to calculate a partition with no more than a predetermined number of clusters by skipping the Expansion step every time the desired number of clusters is reached.
Using the Expansion Step cluster by cluster of Blockloading, the occurrence of empty columns in the matrix $\boldsymbol{Q}^{(ref)}$ is minimized. This saves computational time which would be otherwise wasted for non optimal columns. Blockloading allows the restart with a given partition matrix $\boldsymbol{Q}$ for further improvement. We remark that the greedyICL algorithm of [9] may be used instead of the BlockVB algorithm.

6 Start Value Considerations

The VBEM Algorithm converges to a local optimum [6]. Thus, the choice of the start partition matrix $\boldsymbol{Q}^{(start)}$ affects the outcome of the VBEM algorithm. It takes more iterations of VBEM to converge for a start partition matrix which assigns most of the vertices in a nonoptimal cluster. We also noted during computational tests, that a start partition matrix which assigns a lot of vertices to an non optimal cluster also yields an F which is far away from a (global) optimum. For the Blockloading Algorithm, we need start values for the Expansion and Refinement Step. Except for the first Expansion Step after the Initialization Step, we only need start values for subsets of the vertices for the Blockloading algorithm.

6.1 Ascending Hierarchical Clustering

To select a start partition matrix, an Ascending Hierarchical Clustering Algorithm (AHC) was proposed in the literature (see e.g. [13], [3], [6], [9]).We review the AHC Algorithm explained in the technical documentation of the mixer package [13]. We will explain how to use the AHC Algorithm for our Blockloading algorithm.
For the AHC we need definitions for the distance between vertices and clusters. The distance between two vertices i and j of a directed graph is defined as:

$$d(i,j) = \sum_{k=1}^{N} (A_{ik} - A_{jk})^2 + (A_{ki} - A_{kj})^2 = \|A_i - A_j\|^2. \tag{15}$$

The distance between two clusters is calculated with the help of the barycenters [13]. The barycenters of cluster q for rows and columns, (g_q^+, g_q^-) , with n_q being the number of vertices in cluster q are:

$$g_{qi}^+ = \frac{\sum_{k\in q} A_{ik}}{n_q}, \tag{16}$$

$$g_{qi}^- = \frac{\sum_{k\in q} A_{ki}}{n_q}, \tag{17}$$

for all vertices $i \in \{1,\ldots,N\}$. The distance between two clusters is defined as the Ward distance between the barycenters:

$$\Delta(q,l) = \frac{n_q n_l}{n_q + n_l}\left(\|g_q^+ - g_l^+\|^2 + \|g_q^- - g_l^-\|^2\right). \tag{18}$$

Now, we can recall the AHC of [13] with some minor changes:
(i) *Initialization:* Calculate Δ the distance between the clusters (eqn. (18)). In the first iteration, vertices are considered as clusters, and the distance $d(\cdot,\cdot)$ is used instead.
(ii) *Merging Step:* Merge two clusters if their distance Δ is the smallest. If two distances are equal, clusters to merge are randomly chosen. The label of the new cluster is the smallest of the two previous labels.
(iii) Calculate the distance between groups.
(iv) Iterate (i),(ii) and (iii) until the desired number of clusters, K, is reached.

The AHC algorithm is slow, except for small graphs. It was proposed in [13] to limit the AHC to a subgraph $G^{(sub)}$ with $n_0 < N$ vertices of the original graph. After convergence of the AHC for the n_0 vertices, the other vertices $i \notin G^{(sub)}$ are placed in the clusters with the nearest barycenters.
Another strategy proposed in [13] is to run the k-means algorithm for a user specified number $N_{\max}$ of regularly ordered centers. The centers are $N_{\max}$ vertices of the graph. Then k-means is run until no change of centers occurs. The resulting clusters are used as input for the AHC algorithm.
Now we will explain how we use the AHC for Blockloading. We want to find the start partition matrix in the Expansion Step in Section 5. We run the AHC for all vertices in the active cluster, until we reach $K = 2$ clusters. After the first iterations of Blockloading, the number of vertices in the active cluster is small enough for some of the clusters, so that we can run the AHC without the preparation through the k-means algorithm.
We choose the start partition matrix in the Refinement Step in Section 5 in the following way:
(i) Calculate the barycenters (eqn. (16) and (17)) of the existing clusters including the active cluster.
(ii) Find the shortest distance $d(\cdot,\cdot)$ for each vertex in the active cluster to one of the barycenters. Place the vertex in these clusters of the start partition matrix $\boldsymbol{Q}^{(start)}$ of the Refinement Step.
We also used a second way to initialize the start partition martix $\boldsymbol{Q}^{(start)}$ for the Refinement Step with random entries:
(i) Find all vertices i in the active cluster a.
(ii) Set all entries of the matrix rows $Q_i^{(ref)} \in \mathbb{R}^{1\times K}$ $\forall i \in a$ to zero.
(iii) For each matrix row $i \in a$, draw a randomly initialized vector $Q_i^{(random)}$ with $\sum_{k=1}^K Q_{ik}^{(random)} = 1$ and $Q_{ik}^{(random)} \in (0,1)$ $\forall k = 1,\ldots,K$.

(iv) Set $Q_i^{(ref)} = Q_i^{(random)}\ \forall i \in a$.
The AHC Algorithm provides no procedure for choosing the centers of the k-means preparation. We will address this issue in the next Sections We will propose ways to find the centers (seeds) of k-means based on the properties of the SBM and Blockloading.

6.2 Largest Gap Algorithms

We need a computationally viable algorithm to choose the centers of the k-means algorithm for large graphs with many clusters. This algorithm should provide a fast approximation of the clusters of the SBM. If we choose the centers for k–means or the subnetwork of the AHC randomly, we might end up with many (or in extreme cases all) centers coming from one cluster. The result is a suboptimal initialization of the start value partition matrix. In the following we will show how to lower the probability of such a scenario.
For the case of the undirected Bernoulli SBM with $K = 2$ clusters, an algorithm was introduced in [1] which can find the latent cluster assignments of the vertices of the SBM for graphs of $N = 30$ vertices or more. This algorithm uses the summed degrees of all vertices of the graph. The algorithm was expanded in [12] to the case of more than two clusters and provided an asymptotic criterion for model selection. They called this algorithm the Largest Gaps algorithm (LG algorithm).
We expand the LG algorithm to the directed Bernoulli SBM and the directed Poisson SBM. We calculate the degree $D_i^{(dir)}$ of vertex i for directed graphs as the arithmetic mean of the in–degree $D_i^{(in)}$ and the out–degree $D_i^{(out)}$:

$$D_i^{(in)} = \sum_{j=1}^{N} A_{ij};\ D_i^{(out)} = \sum_{j=1}^{N} A_{ij};\ D_i^{(dir)} = \frac{1}{2}\left(D_i^{(out)} + D_i^{(in)}\right). \tag{19}$$

Now, we define the probability $\overline{\theta}_k^{(dir)} = \frac{1}{2}\left(\sum_{l=1}^{K} \delta_l(\theta_{kl} + \theta_{lk})\right)$ for all clusters $k \in \{1,\ldots,K\}$. Then the degree $D_i^{(dir)}$ is a binomial distributed random variable conditionally on $Z_{ik} = 1$ with parameters $(N-1, \overline{\theta}_k^{(dir)})$. It is $\overline{\theta}_k^{(dir)} \in [0,1]$.
Equivalently, we define for the Poisson SBM
$\overline{\lambda}_k = \frac{1}{2}\left(\sum_{l=1}^{K} \delta_l\left(\lambda_{kl} + \lambda_{lk}\right)\right)$ for all clusters $k \in \{1,\ldots,K\}$. Then, the degree $D_i^{(dir)}$ is a Poisson distributed random variable, conditionally on $Z_{ik} = 1$ with parameter $\overline{\lambda}_k$. It is assumed, that for all clusters $k,l \in \{1,\ldots,K\}$, the separation assumption $k \neq l \rightarrow \overline{\theta}_k \neq \overline{\theta}_l$ holds [12]. We propose a similar assumption for the Poisson SBM: $k \neq l \rightarrow \overline{\lambda}_k \neq \overline{\lambda}_l$ for all $k,l \in \{1,\ldots,K\}$.
The smallest gap between the $\overline{\theta}_k$ ($\overline{\lambda}_k$) is defined as [12]: $\delta = \min_{q \neq r} |\overline{\theta}_q - \overline{\theta}_r|$. It was shown in [12] that a larger gap is to be expected between vertices of different clusters for the Bernoulli SBM. This leads to the LG algorithm. Now we repeat the LG algorithm [12]:

(i) Calculate the vertex degrees $(D_i)_{i=1,\ldots,N}$. Norm the degrees to $T_i = \frac{D_i}{N-1}$.
(ii) Sort the T_i, $i = \{1,\ldots,N\}$ in ascending order:

$$T_i \leq \cdots \leq T_N. \tag{20}$$

(iii) Calculate the $N-1$ gaps, $G_i = T_{i+1} - T_i \forall i \in \{1,\ldots,N-1\}$.
(iv) Find the indices of the $K-1$ largest gaps G_{i_j}: $i_1 < \cdots < i_{K-1}$, such that for all

$k \in \{1,\dots,K-1\}$ and for all $i \in V\backslash\{i_1,\dots,i_{K-1}\}$

$$T_{i_{k+1}} - T_{i_k} \geq T_{i+1} - T_i. \tag{21}$$

(iv) Setting $i_0 = 0$ and $i_K = N$, each index i is assigned to a cluster: $i \to k$ with $i_{k-1} < i \leq i_k$.

We use the vector $(D_i^{(dir)})_{i=1,\dots,N}$ as the input for the directed Bernoulli SBM. The consistency of the LG algorithm was proved for the undirected Bernoulli SBM in [12]. With similar arguments the consistency of the LG algorithm for the directed Bernoulli SBM holds. The prove of the consistency is based on Hoeffding's inequality which only holds for bounded random variables. So, we cannot prove the consistency of the LG algorithm in the case of the Poisson SBM.

The vector of ordered degrees $d = (D_i)_{i=1,\dots,N}$ is called the ordering vector in [1].

The ordered vertex numbers are stored in the vector s.

We found that the gaps provide a grid for choosing the $N_{\max}$ seeds of the AHC algorithm of Section 6.1. Alternatively, we portioned the vector s into k equally spaced intervals and choose a center randomly from each interval. We use the LG algorithm to find the start partition matrix in the Expansion Step of Section 5.

Of course, the LG algorithm can be limited to the vertices in the active cluster.

We want to use the LG algorithm for large graphs, in the first few iterations of Blockloading without the AHC algorithm to save computational time. Therefore we could simply choose the largest gap $\max\limits_{i=1,\dots,N-1} G_i$, and divide the vertices accordingly. Often, the active cluster is divided into two new clusters several times, before an optimum of ICL_{ex} is reached. Choosing the largest gap will not necessarily yield the highest decrease of the ICL_{ex} criterion. Moreover, it was pointed out in [12] that the LG algorithm is suspect to outliers for smaller networks or real world networks. We propose a new algorithm for finding the optimal division of the active cluster in the Expansion step.

We present the **Optimal Gap algorithm** (OG algorithm).

(i) Specify a number of gaps $n_g \ll n_a$. Find the n_g largest gaps $G_{i_1},\dots,G_{i_{n_g}}$.
(ii) For G_{i_j} divide the vertices according to (iv) of the LG algorithm in two clusters.
(iii) Form a start partition matrix $\boldsymbol{Q}_{i_j}^{start}$ where the active cluster is divided according to G_{i_j}.
(iv) Calculate $ICL_{ex_{ij}}$ for $\boldsymbol{Q}_{i_j}^{start}$.
(v) Find $\min\limits_j ICL_{ex_{ij}}$.

We choose the start partition matrix $\boldsymbol{Q}_{i_j}^{(start)}$ corresponding to $\min\limits_j ICL_{ex_{ij}}$. Thus, we have a provably good approximation of the partition matrix near the optimum, with our start value partition matrix $\boldsymbol{Q}_{i_{opt}}^{(start)}$ for $K = 2$ clusters. We avoid outliers by calculating the decrease of the ICL_{ex} for several gaps n_g and not just the largest gap. We remark, that if we apply the OG algorithm repeatedly until no further decrease of the ICL_{ex} is reached, this constitutes a clustering algorithm of its own.

The value of the gaps $\delta_{qr} = |\overline{\theta}_q - \overline{\theta}_r|$, $\forall q \neq r$ varies in most networks. When we run Blockloading (or VBEM in general) for low number of clusters K, we observed that the clusters with the highest δ_{qr} are found first. These results also tend to be more consistent.

6.3 Start Values and the Active Cluster

The output of the LG algorithm is dependent on $\overline{\theta}_k$ (or $\overline{\lambda}_k$). When we build the start partition matrix of the Expansion Step with the LG algorithm, we divide the active cluster in two new clusters k_1 and k_2. If not all connection probabilities θ_{kl} are equal, we have that (without restriction of generality)

$$\hat{\theta}_{k_1} \equiv \sum_{l=1}^{K} \theta_{k_1 l} + \sum_{\substack{l=1 \\ l \neq k_1}}^{K} \theta_{l k_1} > \sum_{l=1}^{K} \theta_{k_2 l} + \sum_{\substack{l=1 \\ l \neq k_2}}^{K} \theta_{l k_2} \equiv \hat{\theta}_{k_2} \tag{22}$$

holds for the summed connection probabilities. Especially for graphs with vectors of summed degrees $(D_i)_{i=1,\ldots,N}$ which have a heavy tail distribution (e.g. a Power Law), the difference between $\hat{\theta}_{k_1}$ and $\hat{\theta}_{k_2}$ will be significant. Vertices which have a low summed degree will only have a few, irregular entries. These vertices are hard to assign to a cluster. We will show how to cluster them as outliers.
If we choose the cluster with the highest $\hat{\theta}_a$ as the active cluster a (see also Section 5), the vertices with low degrees will stay in one cluster for the first iterations of the Blockloading algorithm. This strategy of choosing the active cluster also allows us to impose a threshold T for the cluster connection probabilities. If $\theta_{al} < T$ and $\theta_{la} < T$ for all $l \in \{1,\ldots,K\}$, we can skip the Expansion and Refinement Step for the cluster k_a. Then, after the convergence of all other clusters $k \neq k_a$, we can try to split the cluster in a separate Expansion Step. This allows us to check if any of the two new clusters $k_a^{(1)}$, $k_a^{(2)}$ has connection probabilities higher than the threshold T. Otherwise, this cluster can be left out of the Refinement and Expansion Step. We will compare the different algorithms for finding the start partition matrix in Section 7.

7 Numerical Experiments

For a better comparison of the values of the Free Energy F in the following tests, we take the best of all values of the Free Energy F, F_{ref}, and calculate the difference $\Delta F_{ref} = F - F_{ref} \geq 0$.
To compare different cluster partitions $\boldsymbol{Q}$ and $\boldsymbol{Q}_0$ we use the Normalize Mutual Information (NMI) ([14]). A NMI($\boldsymbol{Q}_0$,$\boldsymbol{Q}$) of 1 means that both partitions $\boldsymbol{Q}$ and $\boldsymbol{Q}_0$ are identical. The NMI is zero when no information about $\boldsymbol{Q}_0$ can be inferred from $\boldsymbol{Q}$.

7.1 Synthetic Networks

7.1.1 Comparison with existing methods

We compare our Blockloading algorithm with existing methods. We reproduced a test performed in [9] for large graphs with a complex structure. The greedyICL algorithm presented in [9], the colsbm algorithm of [15], the vbmod algorithm of [5] and the spectral clustering version of [16] were tested in [9].
The graph is constructed with the Bernoulli SBM of Section 2. There are $N = 10000$ vertices, $K = 50$ clusters. The probabilities for the cluster assignments are given by the vector $\boldsymbol{\delta} = (1/50,\ldots,1/50) \in \mathbb{R}^{1\times 50}$. The probabilities for the existence of an edge are generated according to

$$\theta_{kl} = \begin{cases} ZU + (1-Z)\varepsilon, \text{ if } k \neq l \\ U, \text{ if } k = l \end{cases} \tag{23}$$

with $Z \sim \text{Bernoulli}(0,1)$, the uniform distribution $U \sim \mathscr{U}(0.45)$ and $\varepsilon = 0.01$.
[9] applied their greedyICL algorithm to 20 simulated graphs. They found that their greedyICL algorithm outperformed the other algorithms with an average NMI of 0.88 between the real cluster partition and the calculated one. The greedyICL returned $K = 80$ clusters most of the time [9].
We applied our Blockloading algorithm for the Bernoulli SBM to 20 simulated graphs. We used the AHC algorithm of Section 6.1 with a subnetwork for the Expansion Step and and a random sub matrix for the Refinement Step. The active cluster was chosen as the cluster with the most vertices (see Section 5).
We found that our Blockloading algorithm identified the correct number of clusters, $K = 50$ and the correct partition with a NMI of 1.0 (each vertex was placed in the correct cluster) for each of the 20 runs.
We conclude that our Blockloading algorithm outperforms the other algorithms mentioned above in this test. The Blockloading algorithm is the only algorithm of the mentioned algorithms in this Section, which is able to identify the correct number of clusters.

7.1.2 Large and complex Poisson SBM

To test the Blockloading algorithm for the Poisson SBM we constructed the following example graph which was generated as explained in Section 2. Each generated graph has $N = 10\,000$ vertices and $K = 50$ clusters. The vector of the Multinomial distribution is given by $\boldsymbol{\delta} = (1/50, \ldots, 1/50) \in \mathbb{R}^{1\times 50}$. We draw $Z \sim \text{Bernoulli}(0.2)$ and set $\varepsilon = 1$. The rates of the edge weights λ_{kk} within the clusters are set in ascending order to $(0.1, 0.2, \ldots, 1, 1.5, 2, \ldots, 21) \in \mathbb{R}^{1\times 50}$. We set the parameters $\lambda_{k\cdot}$ according to

$$\lambda_{kl} = \begin{cases} Z\lambda_{kk} + (1-Z)\varepsilon, \text{ if } k \neq l \\ \lambda_{kk}, \text{ if } k = l \end{cases}. \tag{24}$$

The generated graph has an asymmetric structure and varying edge weights.
Again, we used the AHC algorithm for the Expansion step and the random initialization for the Refinement Step.
We generated 20 realisations of the graph and ran the Blockloading algorithm for the Poisson SBM. We found that Blockloading returned the correct number of clusters for each test. We calculated a NMI of 1.0 for each comparison between the calculated partition matrix $\boldsymbol{Q}$ and the true partition matrix $\boldsymbol{Q}_{true}$.
This example shows, that our Blockloading algorithm is able to reliably calculate correct results for large weighted graphs with complex structure.

7.2 Southern California Earthquake Network

The Earthquake Network (EN) was introduced in [8]. It was shown in ([8, 17, 18]) that important statistical properties of earthquake activity are inherited by the EN.
The EN is constructed for a chosen geographical area and time span. A square grid is put on the area of interest [19]. The EN unfolds in the following way:
(i) Place a vertex in the first square where seismic activity occurs at the start of the observation interval.
(ii) Place a second vertex where the next time seismic activity occurs and place a (directed) edge between the last two vertices of seismic activity pointing to the latest vertex of activity.

(iii) Continue until the end of observation.

For the Southern California area $(32s, 37n; 122w, 114w)$ we constructed the Earthquake Network for the time interval from January 1, 1984 to December 31, 2013. We chose a square length of $10\,km$ for the grid. This results in 4256 squares. We used the earthquake catalogue data from the Southern California Earthquake Data center (SCDEC) [20].
Earthquake catalogues have a minimum magnitude of completeness (see e.g. [21]). The earthquake catalogue is expected to list every earthquake with magnitude equal or higher than the magnitude of completeness. It was shown in [21] that the SCDEC catalogue is complete for a magnitude of $M \geq 1.8$ on the Richter Scale from January 1, 1984 onwards. We used only earthquakes with magnitude $M \geq 1.8$ for the construction of the EN.
We set the entries on the diagonal of the adjacency matrix of the EN to zero. These entries represent aftershocks in the EN. The resulting adjacency matrix of the graph of the EN has $N = 2324$ vertices and 58 718 edges.
We found that the directness of the Earthquake Network, which results from the construction as explained above, is an important feature of earthquake networks. To measure the importance of the directedness of the network we used the degree of asymmetry (DOA) [22]. It allows to quantify the directedness of a network. A DOA of 0 means no directedness whereas 1 stands for a perfectly directed network. We found that the DOA of the ENSC introduced above, was 0.86.
We ran our Blockloading algorithm with the Bernoulli SBM and the Poisson SBM for ten times for the EN. To test the Bernoulli SBM, we set each edge weight bigger than zero of the adjacency matrix to one. We also tested the VBEM–ILvb [6] algorithm of the Bernoulli SBM mentioned in Section 3. We initialized the VBEM–ILvb algorithm for 1 to 15 clusters and took the best result measured by the lowest Free Energy.
We implemented all algorithms and used the AHC algorithm of Section 6.1 with a subnetwork for the Expansion Step and and a random sub matrix for the Refinement Step. We used the strategy outlined in Section 6.3 and chose the cluster with the highest $\hat{\theta}_k$ (or $\hat{\lambda}_k$ for the Poisson SBM). We also tried choosing the cluster with the highest vertex count as the active cluster. This produced clearly inferior results (not shown here) due to sparsity of the network and the heavy tail distribution of the vertex degrees.
We ran VBEM–ILvb, initialized for 1 to 15 clusters, and Blockloading for 10 times. The best result measured by the lowest Free Energy was obtained by Blockloading with $F = 185\,812$, ($\Delta F_{ref} = 0$) and $K^* = 12$ clusters. This result is shown in Fig. 1. The clustering result in Fig. 1 shows, that the areas in the same cluster are spread over the area and are not necessarily geographically adjacent because they depend on edge connection profiles.
The worst result of Blockloading was $\Delta F_{ref} = 5$ and $K = 12$. The comparison of the partition matrices of the best result and the worst result yields a NMI of 0.96. We also ran Blockloading 10 times without the Refinement Step. The results varied between $\Delta F_{ref} = 109$, with a NMI of 0.77, $K = 16$ clusters and $\Delta F_{ref} = 248$, with a NMI of 0.77 and also $K = 16$ clusters.
The best result of VBEM–ILvb was, measured by the difference to the reference Free Energy, $\Delta F_{ref} = 350$ with $K = 11$ clusters, and the worst result was $\Delta F_{ref} = 1\,323$ with $K = 9$. We compared all partition matrices in terms of NMI with the best partition in Table 1 and displayed the calculated number of clusters. We conclude that Blockloading has substantially higher consistency of the results. The computational speed of the best result of Blockloading was 10 times faster (52 times faster without the Refinement

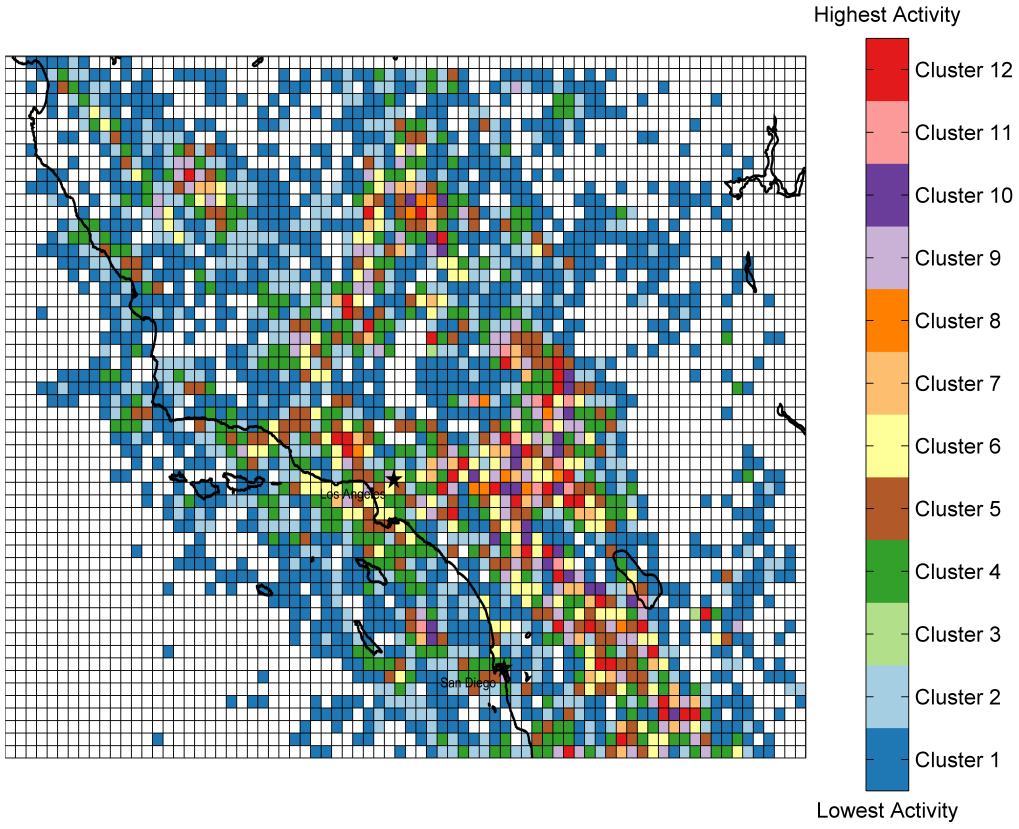

Figure 1: The best result of the Blockloading algorithm for the Bernoulli SBM for the Southern California Earthquake Network is shown. All areas in the same cluster are colored in the same color. Areas with no seismic activity in the observation time are in white. The coastoutline was plotted with [23].

Table 1: Results of the Blockloading algorithm for the Bernoulli SBM of the Earthquake Network. Normalized Mutual Information (NMI) calculated in comparison with the best result of all tests. Results were ordered according to quality of NMI. Number of clusters K. Difference to reference Free Energy ΔF_{ref}.

NMI	1	0.99	0.99	0.99	0.98	0.98	0.97	0.97	0.96	0.96
K	12	12	12	12	12	12	12	12	12	12
ΔF_{ref}	0	0.8	0.1	1.1	2	0.1	0.8	2.1	5.2	4.2

Table 2: VBEM–ILvb results for the Earthquake Network. Normalized Mutual Information (NMI) calculated in comparison to the best result of all tests. Results were ordered according to quality of NMI. Number of clusters K. Difference to reference Free Energy ΔF_{ref}.

NMI	0.79	0.77	0.77	0.77	0.76	0.76	0.76	0.76	0.75	0.75
K	14	10	10	10	14	8	11	15	10	9
ΔF_{ref}	705	365	365	362	374	715	350	693	1337	1323

Table 3: Results of the Blockloading algorithm for the Poisson SBM for the weighted Earthquake Network. Normalized Mutual Information (NMI) calculated in comparison to the best result of all tests. Results were ordered according to quality of NMI. Number of clusters K. Difference to reference Free Energy ΔF_{ref}.

NMI	1	0.92	0.89	0.87	0.88	0.90	0.91	0.75	0.91	0.91
K	46	46	46	48	48	48	48	45	44	46
ΔF_{ref}	0	20	132	163	198	228	276	304	346	517

Step) than the best result of VBEM–ILvb on the same machine.
We tested the Blockloading algorithm for the Poisson SBM for the for the weighted Earthquake Network. The highest edge weight was 240 and the lowest 1. We ran the Blockloading algorithm for ten times. The best result measured by the Free Energy F yielded $K = 45$ clusters and the worst result $K = 46$ clusters with $\Delta F_{ref} = 517$. The results are shown in table 3.
We note that the edge weights provide a lot of additional information. In the best clustering result, 9 for 45 clusters are clusters with only one vertex. These nine vertices are hubs which are strongly linked to all clusters in the network. Hubs play an important role in Earthquake Networks [8]. We found during testing, that these hubs can only reliably identified with our Blockloading algorithm. The weighted graph is also harder to cluster, because the results vary more than for the unweighted graph.

8 Conclusion

We developed a new algorithm for applying Variational Bayesian (VB) methods to large and complex graphs. The task of finding the optimal number of clusters was combined with the calculation of the cluster partition matrix, showing that both tasks are essentially linked for VB methods. The consistency of the results was substantially improved without the need for an initial guess of the numbers of clusters. Our method was tested on a synthetic network drawn from an SBM and an Earthquake Network. In those tests, every run of our algorithm performed substantially better than the best result of the comparable algorithms for solving the SBM in terms of computational speed and quality of the results.

Acknowledgements

C.T.W. acknowledges the funding through a GeoSim fellowship.

A Bernoulli BlockVB Algorithm

Input: partition matrix $Q^{(start)}$, active Cluster c and adjacency matrix A.
Initialization: Find indices I of vertices in the active cluster, $i \in c$.
Initialize parameters $\alpha_{kl}^0 = 1/2$, $\beta_{kl}^0 = 1/2\ \forall k,l \in \{1,\dots,K\}$ for the Beta prior distributions and $\delta_k^0 = 1\,\forall k \in \{1,\dots,K\}$ for the Dirichlet prior distribution with non informative priors [24, 6].
To save computational time in the M–Step, we prepare the update formulas for the Maximization Step dependent on the vertices in $i \in I$: $S_{\alpha_{xy}} = \sum_{i\neq j}^{N} Q_{ix}Q_{jy}A_{ij}$, $S_{\beta_{xy}} = \sum_{i\neq j}^{N} Q_{ix}Q_{jy}(1-A_{ij})$, $S_{\delta_k} = \sum_{i=1}^{N} Q_{ik}$. Calculate

$$S^I_{\alpha_{xy}} = \sum_{\substack{i,j\in I\\ i\neq j}} Q_{ix}Q_{jy}A_{ij} + \sum_{\substack{i\in I\\ i\neq j}}\sum_{\substack{j\notin I\\ j=1}}^{N} Q_{ik}Q_{jl}A_{ij} + \sum_{\substack{j\in I\\ i\neq j}}\sum_{\substack{i\notin I\\ i=1}}^{N} Q_{ik}Q_{jl}A_{ij}, \tag{25}$$

$$S^I_{\beta_{xy}} = \sum_{\substack{i,j\in I\\ i\neq j}} Q_{ix}Q_{jy}(1-A_{ij}) + \sum_{\substack{i\in I\\ i\neq j}}\sum_{\substack{j\notin I\\ j=1}}^{N} Q_{ik}Q_{jl}(1-A_{ij}) + \sum_{\substack{j\in I\\ i\neq j}}\sum_{\substack{i\notin I\\ i=1}}^{N} Q_{ik}Q_{jl}(1-A_{ij}), \tag{26}$$

$$S^I_{\delta_k} = \sum_{i\in I} Q_{ik}. \tag{27}$$

Calculate the update of the parameters:

$$\alpha_{xy} = S_{\alpha_{xy}} + \alpha^0_{xy}, \tag{28}$$

$$\beta_{xy} = S_{\beta_{xy}} + \beta^0_{xy}, \tag{29}$$

$$\delta_x = \sum_{i=1}^{N} Q_{ix} + \delta^0_x. \tag{30}$$

$$\tag{31}$$

Main Loop: Until convergence of the Free Energy $F[q(\cdot)]$:
Expectation Step: Until convergence of the matrix entries Q_{ik}, $\forall i \in I$, $k = \{1,\dots,K\}$.
Update of the matrix entries Q_{lv} for all $l \in I$ and for all $v \in \{1,\dots,K\}$:

$$Q_{lv} \propto \exp\left(-\sum_{y=1}^{K}\sum_{\substack{j\in I\\ j\neq l}} A_{lj}Q_{vy}C_{vy} - \sum_{y=1}^{K}\sum_{\substack{j\in I\\ j\neq l}} A_{jl}Q_{jy}C_{yv} + D_{vy}\sum_{\substack{j\in I\\ j\neq l}} Q_{jy} \right), \tag{32}$$

where $C_{vy} = \psi(\beta_{vy}) - \psi(\alpha_{vy})$ and $D_{vy} = \psi(\beta_{vy}) - \psi(\beta_{yv}) - \psi(\alpha_{vy}+\beta_{vy}) - \psi(\alpha_{yv}+\beta_{yv})$. Calculate the norm $Q^{\star}_{ik} = Q_{ik} / \left(\sum_{k=1}^{K} Q_{ik}\right)$ of the updates Q_{ik}.
Check for convergence of the matrix entries Q_{ik}, $\forall i \in I$, $k = 1,\dots,K$.
Maximization Step (M–Step):
Update the parameters of variational distributions. The parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of the Beta distributions and the parameters $\boldsymbol{\delta}$ of the Dirichlet distribution: $S^{I_{old}}_{\alpha_{xy}} = S^I_{\alpha_{xy}}$, $S^{I_{old}}_{\beta_{xy}} = S^I_{\beta_{xy}}$, $S^{I_{old}}_{\delta_k} = S^I_{\delta_k}$. Calculate $S^I_{\alpha_{xy}}$, $S^I_{\beta_{xy}}$ and $S^I_{\delta_k}$. Do the updates $\alpha_{xy} = S_{\alpha_{xy}} - S^{I_{old}}_{\alpha_{xy}} + S^I_{\alpha_{xy}} + \alpha^0_{xy}$, $\beta_{xy} = S_{\beta_{xy}} - S^{I_{old}}_{\beta_{xy}} + S^I_{\beta_{xy}} + \beta^0_{xy}$ and $\delta_k = S_{\delta_k} - S^{I_{old}}_{\delta_k} + S^I_{\delta_k} + \delta^0_k$.

Calculate the Free Energy F:

$$F\left[\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\delta}\right] = \ln\left(\frac{\Gamma\left(\sum_{x=1}^{K}\delta_x\right)\prod_{x=1}^{K}\Gamma(\delta_x^0)}{\Gamma\left(\sum_{x=1}^{K}\delta_x^0\right)\prod_{x=1}^{K}\Gamma(\delta_x)}\right) + \sum_{x,y}^{K}\ln\left(\frac{B(\alpha_{xy},\beta_{xy})}{B(\alpha_{xy}^0,\beta_{xy}^0)}\right) + \sum_{i=1}^{N}\sum_{x=1}^{K} Q_{ix}\ln Q_{ix}. \tag{33}$$

Check for convergence of F.

B Poisson Stochastic Block Model

Proposition 2. *The optimal estimate of the expectation of the latent variable Z_{ik}, $\mathbb{E}[z_{ik}] = Q_{ik}$ for all $i \in V, q = 1,\ldots,N$, $Q_{iv}^{\star} = \arg\min_{Q_{iv}} F(\boldsymbol{Q},\boldsymbol{\vartheta})$, is given by:*

$$Q_{iv} \propto \exp\Big(\sum_{\substack{i=1\\i\neq j}}^{N}\sum_{k=1}^{K} A_{ai}Q_{ik}C_{vk} + \sum_{\substack{i=1\\i\neq j}}^{N}\sum_{k=1}^{K} A_{ia}Q_{ik}C_{vk} \tag{34}$$

$$- \sum_{\substack{i=1\\i\neq j}}^{N}\sum_{k=1}^{K} Q_{ik}D_{vk} + G_v\Big), \tag{35}$$

where $\mathbb{E}_{\boldsymbol{\lambda}}[\log\lambda_{vk}] = \psi(\alpha_{vk}) - \log(\beta_{vk}) = C_{vk}$, $\mathbb{E}_{\boldsymbol{\lambda}}[\lambda_{vk}] + \mathbb{E}_{\boldsymbol{\lambda}}[\lambda_{kv}] = \frac{\alpha_{vk}}{\beta_{vk}} + \frac{\alpha_{kv}}{\beta_{kv}} = D_{vk}$, $\mathbb{E}_{\boldsymbol{\delta}}[\boldsymbol{\delta}_q] = \psi(\delta_q) - \psi\big(\sum\big(\sum_{l=1}^{K}\delta_l\big)\big) = G_q$ and $\psi(\cdot)$ is the Digamma function.

Proof. The terms of the lower bound F dependent on $\boldsymbol{Q}$ are:

$$F\left[\boldsymbol{Q}\right] = \sum_{q,l}^{K}\sum_{\substack{i,j\\i\neq j}}^{N}\big(-Q_{iq}Q_{jl}A_{ij}\big(\psi(\alpha_{ql}) - \ln(\beta_{ql})\big) + Q_{iq}Q_{jl}\left(\frac{\alpha_{ql}}{\beta_{ql}}\right)\big) - \sum_{i=1}^{N}\sum_{q=1}^{N} Q_{iq}\left(\psi(\delta_q) - \psi\Big(\sum\left(\sum_{l=1}^{K}\delta_l\right)\Big)\right) + \sum_{i=1}^{N}\sum_{k=1}^{K} Q_{ik}\ln Q_{ik} + const.. \tag{36}$$

We take the derivative of F with respect to Q_{iv}:

$$\frac{\partial F\left[\boldsymbol{Q}\right]}{\partial Q_{iv}} = -\sum_{\substack{i=1\\i\neq a}}^{N}\sum_{q=1}^{K} A_{ai}Q_{iq}C_{vq} + \sum_{\substack{i=1\\i\neq a}}^{N}\sum_{q=1}^{K} Q_{iq}D_{vq} - G_v + \ln Q_{av}. \tag{37}$$

We set eqn. 37 to zero. This yields:

$$Q_{iv} \propto \exp\Big(\sum_{\substack{i=1\\i\neq j}}^{N}\sum_{q=1}^{K} A_{ai}Q_{iq}C_{vq} + \sum_{\substack{i=1\\i\neq j}}^{N}\sum_{q=1}^{K} A_{ia}Q_{iq}C_{qv} \tag{38}$$

$$- \sum_{\substack{i=1\\i\neq j}}^{N}\sum_{q=1}^{K} Q_{iq}D_{vq} + G_v\Big). \tag{39}$$

□

The parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta})$ of the conjugate prior distributions are updated in the Maximization Step (M-Step), according to:

Proposition 3. *The optimization of the variational bound of eqn. 8 for $q(\lambda_{kl})$ for all $k,l = 1,\ldots,K$ shows, that $q(\lambda_{kl})$ has the functional form of a $\Gamma(\lambda_{kl};\alpha_{kl},\beta_{kl})$ distribution. It has the same functional form as the prior distribution $p(\lambda_{kl}^0) = \Gamma(\lambda_{kl};\alpha_{kl}^0,\beta_{kl}^0)$. The hyperparameters α_{kl} and β_{kl} for all $k,l = 1,\ldots,K$ for the partition matrix $\boldsymbol{Q}$ are:*

$$\alpha_{kl} = \sum_{i\neq j}^{N} Q_{ik}Q_{jl}A_{ij} + \alpha_{kl}^0, \tag{40}$$

$$\beta_{kl} = \sum_{i\neq j}^{N} Q_{ik}Q_{jl} + \beta_{kl}^0. \tag{41}$$

Proof. The terms of the lower bound F dependent on $\boldsymbol{\lambda}$ are:

$$\begin{aligned} F\left[q(\boldsymbol{\lambda})\right] = &\sum_{q,l}^{K}\sum_{\substack{i,j\\ i\neq j}}^{N}\left(-Q_{iq}Q_{jl}A_{ij}\mathbb{E}_{\boldsymbol{\lambda}}\left[\ln\lambda_{ql}\right] + Q_{iq}Q_{jl}\mathbb{E}_{\boldsymbol{\lambda}}\left[\lambda_{ql}\right]\right) \\ &- \mathbb{E}_{\boldsymbol{\lambda}}\left[\ln p(\boldsymbol{\lambda})\right] + \mathbb{E}_{\boldsymbol{\lambda}}\left[\ln q(\boldsymbol{\lambda})\right] + const. \end{aligned} \tag{42}$$

We take the variational derivative of F which yields:

$$\begin{aligned} \frac{\delta F\left[q(\cdot)\right]}{q(\lambda_{ql})} = &\sum_{\substack{i,j\\ i\neq j}}^{N}\left(-Q_{iq}Q_{jl}A_{ij}\ln\lambda_{ql} + Q_{iq}Q_{jl}\lambda_{ql}\right) + \ln q(\lambda_{ql}) \\ &\left(\alpha_{ql}^0\ln(\beta_{ql}^0) - \ln\Gamma(\alpha_{ql}^0) + (\alpha_{ql}^0 - 1)\ln\lambda_{ql} - \beta_{ql}^0\lambda_{ql}\right). \end{aligned} \tag{43}$$

It follows that

$$q(\lambda_{ql}) \propto \exp\Big(\big(\sum_{\substack{i,j\\ i\neq j}}^{N} Q_{iq}Q_{jl}A_{ij} + \alpha_{ql}^0 - 1\big)\ln\lambda_{ql} - \big(\sum_{\substack{i,j\\ i\neq j}}^{N} Q_{iq}Q_{jl} + \beta_{ql}^0\big)\lambda_{ql}\Big). \tag{44}$$

Equation 44 shows that $q(\lambda_{ql})$ has the functional form of a $\Gamma(\sum_{i\neq j}^{N} Q_{ik}Q_{jl}A_{ij} + \alpha_{kl}^0, \sum_{i\neq j}^{N} Q_{ik}Q_{jl} + \beta_{kl}^0)$ distribution.

□

Proposition 4 ([6])**.** *The optimization of the Free Energy (eqn. 8) for $q(\delta_k)$ for all $k = 1,\ldots,K$, shows that $q(\delta_k)$ has the functional form of a Dirichlet $Dir(\boldsymbol{\delta};\boldsymbol{\delta})$ distribution. The hyper parameters δ_k for all $k = 1,\ldots,K$ are*

$$\delta_k = \sum_{i=1}^{K} Q_{ik} + \delta_k^0, \tag{45}$$

where $\boldsymbol{Q}$ is the partition matrix.

Proposition 5. *The Free Energy after convergence (ILvb) for the Poisson Stochastic Block Model for K cluster, partition matrix $\boldsymbol{Q}$ and parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta})$ is given by:*

$$\begin{aligned} F[\boldsymbol{Q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\delta}] = &\sum_{k,l}^{K}\ln\left(\frac{\beta_{kl}^{\alpha_{kl}}\Gamma(\alpha_{kl}^0)}{{\beta_{kl}^0}^{\alpha_{kl}^0}\Gamma(\alpha_{kl})}\right) + \sum_{i=1}^{N}\sum_{k=1}^{K} Q_{ik}\ln Q_{ik} \\ &+ \ln\left(\frac{\Gamma\left(\sum_{x=1}^{K}\delta_x\right)\prod_{x=1}^{K}\Gamma(\delta_x^0)}{\Gamma\left(\sum_{x=1}^{K}\delta_x^0\right)\prod_{x=1}^{K}\Gamma(\delta_x)}\right). \end{aligned} \tag{46}$$

Proof. The Free Energy (Variatioal Lower bound) in eqn. 6 is given by:

$$F[q(\cdot)] = -\sum_{\boldsymbol{Z}} \int \ln\left(\frac{p(\boldsymbol{A},\boldsymbol{Z}|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})}{q(\boldsymbol{Z},\boldsymbol{\vartheta})}\right) q(\boldsymbol{Z},\boldsymbol{\vartheta})\mathrm{d}\boldsymbol{\vartheta} \tag{47}$$

$$\begin{aligned}
&= -\mathbb{E}_{\boldsymbol{Z},\boldsymbol{\lambda},\boldsymbol{\delta}}[\ln p(\boldsymbol{A}|\boldsymbol{Z},\boldsymbol{\vartheta})] - \mathbb{E}_{\boldsymbol{\lambda},\boldsymbol{\delta}}[\ln p(\boldsymbol{Z}|\boldsymbol{\delta})] - \mathbb{E}_{\boldsymbol{\lambda}}[\ln p(\boldsymbol{\lambda})] \\
&- \mathbb{E}_{\boldsymbol{\delta}}[\ln \boldsymbol{\delta}] + \sum_{i=1}^{N} \mathbb{E}_{Z_i}[\ln q(Z_i)] + \mathbb{E}_{\boldsymbol{\delta}}[\ln q(\boldsymbol{\delta})] + \mathbb{E}_{\boldsymbol{\lambda}}[\ln q(\boldsymbol{\lambda})] \qquad (48) \\
&= \sum_{q,l}^{K} \sum_{\substack{i,j \\ i\neq j}}^{N} \big(- Q_{iq}Q_{jl}A_{ij}\left(\psi(\alpha_{ql}) - \ln(\beta_{ql})\right) + Q_{iq}Q_{jl}\left(\frac{\alpha_{ql}}{\beta_{ql}}\right)\big) \\
&+ \sum_{q,l}^{K} -(\alpha_{ql}^0 - 1)(\psi(\alpha_{ql}) - \ln(\beta_{ql})) + \ln\left(\Gamma(\alpha_{ql}^0\right) \\
&+ \beta_{ql}^0\left(\frac{\alpha_{ql}}{\beta_{ql}}\right) - \alpha_{ql}^0 \ln(\beta_{ql}^0) + \alpha_{ql}\ln(\beta_{ql}) - \ln\left(\Gamma(\alpha_{ql})\right) \\
&+ (\alpha_{ql} - 1)\left(\psi(\alpha_{ql}) - \ln(\beta_{ql})\right) - \beta_{ql}\left(\frac{\alpha_{ql}}{\beta_{ql}}\right) \\
&- \sum_{q=1}^{K}\sum_{i=1}^{N} Q_{iq}\big(\psi(\delta_q) - \psi\left(\sum_{l=1}^{K}\delta_q\right)\big) - \ln\left(\Gamma\left(\sum_{l=1}^{K}\delta_l^0\right)\right) \\
&+ \sum_{q=1}^{K}\ln\left(\Gamma(\delta_q^0)\right) - \sum_{q=1}^{K}(\delta_q^0 - 1)\left(\psi(\delta_q) - \psi\left(\sum_{l=1}^{K}\delta_l\right)\right) \\
&+ \ln\left(\Gamma\left(\sum_{q=1}^{K}\delta_q\right)\right) - \sum_{q=1}^{K}\ln\left(\Gamma(\delta_q)\right) \\
&+ \sum_{q=1}^{K}(\delta_q - 1)\left(\psi(\delta_q) - \psi\left(\sum_{l=1}^{K}\delta_l\right)\right) + \sum_{i=1}^{N}\sum_{k=1}^{K} Q_{ik}\ln Q_{ik}. \qquad (49)
\end{aligned}$$

We plug the Update Equations of Propositions 3 and 4 for α_{kl}, β_{kl} and $\delta_k \forall k,l \in \{1,\ldots,K\}$ into eqn. 49. This yields eqn. 46.

□

B.1 Poisson Block VB algorithm

Input: partition matrix $Q^{(start)}$, active Cluster c and adjacency matrix A.
Initialization: Find indices I of vertices in the active cluster, $i \in c$.
Set non informative prior parameters for the Gamma prior distribution: $\alpha_{kl}^0 = \frac{1}{3}$ and $\beta_{kl}^0 = 1/100$ for all k,l [25], and for the Dirichlet distributions $\delta_k^0 = 1\,\forall k$ [6, 24].
Initialize update formulas for the M-Step:

$$\begin{aligned}
S_{\alpha_{kl}} &= \sum_{i\neq j}^{N} Q_{ik}Q_{jl}A_{ij}, \\
S_{\beta_{kl}} &= \sum_{i\neq j}^{N} Q_{ik}Q_{jl}, \\
S_{\delta_k} &= \sum_{i=1}^{N} Q_{ik}
\end{aligned}$$

Prepare

$$S^I_{\alpha_{kl}} = \sum_{\substack{i,j\in I\\ i\neq j}} Q_{ik}Q_{jl}A_{ij} + \sum_{\substack{i\in I\\ i\neq j}}\sum_{\substack{j\notin I\\ j=1}}^{N} Q_{ik}Q_{jl}A_{ij} + \sum_{\substack{j\in I\\ i\neq j}}\sum_{\substack{i\notin I\\ i=1}}^{N} Q_{ik}Q_{jl}A_{ij}, \tag{50}$$

$$S^I_{\beta_{kl}} = \sum_{\substack{i,j\in I\\ i\neq j}} Q_{ik}Q_{jl} + \sum_{\substack{i\in I\\ i\neq j}}\sum_{\substack{j\notin I\\ j=1}}^{N} Q_{ik}Q_{jl} + \sum_{\substack{j\in I\\ i\neq j}}\sum_{\substack{i\notin I\\ i=1}}^{N} Q_{ik}Q_{jl}, \tag{51}$$

$$S^I_{\delta_k} = \sum_{i\in I} Q_{ik}. \tag{52}$$

Calculate the parameters $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\delta})$ $\forall k, l \in \{1,\ldots,K\}$ according to Proposition 3 and 4: $\alpha_{kl} = S_{\alpha_{kl}} + \alpha^0_{kl}$, $\beta_{kl} = S_{\beta_{kl}} + \beta^0_{kl}$, $\delta_{kl} = S_{\delta_k} + \delta^0_k$.
Main Loop: Until convergence of F.
Expectation Step: Until convergence of the matrix entries Q_{ik}, $\forall i \in I$, $k = \{1,\ldots,K\}$.
Calculate updates of all matrix entries Q_{ik} $i \in I$ and $k = 1,\ldots,K$ according to Proposition 2.
Calculate the norm $Q^\star_{ik} = Q_{ik} / \left(\sum_{k=1}^K Q_{ik}\right)$ of the updates Q_{ik}.
Check for convergence of the matrix entries Q_{ik}, $\forall i \in I$, $k = 1,\ldots,K$.
Maximization Step: Update the parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ of the Gamma prior distributions and the parameters $\boldsymbol{\delta}$ of the Dirichlet prior distributions. Set $S^{old}_{\alpha_{kl}} = S^I_{\alpha_{kl}}$, $S^{old}_{\beta_{kl}} = S^I_{\beta_{kl}}$ and $S^{old}_{\delta_k} = S^I_{\delta_k}$. Calculate $S^I_{\alpha_{kl}}$, $S^I_{\beta_{kl}}$ and $S^I_{\delta_k}$. Calculate M–Step Updates: $\alpha_{kl} = S_{\alpha_{kl}} - S^{old}_{\alpha_{kl}} + S^I_{\alpha_{kl}} + \alpha^0_{kl}$, $\beta_{kl} = S_{\beta_{kl}} - S^{old}_{\beta_{kl}} + S^I_{\beta_{kl}} + \beta^0_{kl}$, $\delta_k = S_{\delta_k} - S^{old}_{\delta_k} + S^I_{\delta_k} + \delta^0_k$.
Calculate $F(\boldsymbol{Q},\boldsymbol{\vartheta})$ according to Proposition 5. Check for convergence of F.

C Exact Integrated Classification Likelihood Criterion

Proof of the ICL_{ex} for the Poisson SBM presented in Proposition 1:

Proof. We assumed factorized prior distributions
$p(\boldsymbol{\lambda}) = \prod_{k,l}^K \text{Gamma}(\lambda_{kl}; \alpha^0_{kl}, \beta^0_{kl})$
and $p(\boldsymbol{\delta}) = \text{Dir}(\boldsymbol{\pi}; \boldsymbol{\delta}^0 = (\delta^0_1,\ldots,\delta^0_k))$. Thus, the integrated complete data log likelihood can written in the following way [9]:

$$\ln p(\boldsymbol{A},\boldsymbol{Z}|K) = \ln\left(\int_{\boldsymbol{\pi},\boldsymbol{\lambda}} p(\boldsymbol{A},\boldsymbol{Z},\boldsymbol{\pi}|K)\mathrm{d}\boldsymbol{\pi}\mathrm{d}\boldsymbol{\lambda}\right) \tag{53}$$

$$= \ln\Big(\int_{\boldsymbol{\lambda}} p(\boldsymbol{A}|\boldsymbol{Z},\boldsymbol{\lambda},K)p(\boldsymbol{\lambda}|K)\mathrm{d}\boldsymbol{\lambda}\int_{\boldsymbol{\pi}} p(\boldsymbol{Z}|\boldsymbol{\pi},K)p(\boldsymbol{\pi}|K)\mathrm{d}\boldsymbol{\pi}\Big) \tag{54}$$

$$= \ln p(\boldsymbol{A}|\boldsymbol{Z},K) + \ln p(\boldsymbol{Z}|K). \tag{55}$$

The term $p(\boldsymbol{A}|\boldsymbol{Z},K)$ is together with eqn. (3):

$$p(\boldsymbol{A}|\boldsymbol{Z},K) = \int_{\boldsymbol{\lambda}} p(\boldsymbol{A}|\boldsymbol{Z},\boldsymbol{\lambda},K)p(\boldsymbol{\lambda}|K)\mathrm{d}\boldsymbol{\lambda} \tag{56}$$

$$= \int_{\boldsymbol{\lambda}} \left(\prod_{k,l}^{K} \frac{\lambda_{kl}^{\sum_{i\neq j}^{N} Z_{ik}Z_{jl}A_{ij}}}{(A_{ij}!)^{\sum_{i\neq j}^{N} Z_{ik}Z_{jl}}} \exp\left(-\lambda_{kl} \left(\sum_{i\neq j}^{N} Z_{ik}Z_{jl} \right) \right) \right)$$

$$\times \prod_{k,l}^{K} \frac{{\beta_{kl}^{0}}^{\alpha_{kl}^{0}}}{\Gamma(\alpha_{kl}^{0})} \lambda_{kl}^{\alpha_{kl}^{0}-1} e^{-\beta_{kl}^{0}\lambda_{kl}} \mathrm{d}\boldsymbol{\lambda} \tag{57}$$

$$= \int_{\boldsymbol{\lambda}} \prod_{k,l}^{K} \frac{{\beta_{kl}^{0}}^{\alpha_{kl}^{0}}}{\Gamma(\alpha_{kl}^{0})} \lambda_{kl}^{\sum_{i\neq j}^{N} Z_{ik}Z_{jl}A_{ij}+\alpha_{kl}^{0}-1} \frac{1}{(A_{ij}!)^{\sum_{i\neq j}^{N} Z_{ik}Z_{jl}}}$$

$$\exp\left(-\left(\sum_{i\neq j} Z_{ik}Z_{jl} + \beta_{kl}^{0} \right) \lambda_{kl} \right) \mathrm{d}\boldsymbol{\lambda} \tag{58}$$

$$= \prod_{k,l}^{K} \frac{\Gamma(\alpha_{kl})}{\Gamma(\alpha_{kl}^{0})} \frac{{\beta_{kl}^{0}}^{\alpha_{kl}^{0}}}{\beta_{kl}^{\alpha_{kl}}} \frac{1}{(A_{ij}!)^{\sum_{i\neq j}^{N} Z_{ik}Z_{jl}}} \tag{59}$$

In [9] it was shown that the term $p(\boldsymbol{Z}|K)$ is:

$$p(\boldsymbol{Z}|K) = \int_{\boldsymbol{\pi}} p(\boldsymbol{Z}|\boldsymbol{\pi},K)p(\boldsymbol{\pi}|K)\mathrm{d}\boldsymbol{\pi} \tag{60}$$

$$= \int_{\boldsymbol{\pi}} \left(\prod_{k=1}^{K} \pi_k^{\sum_{i=1}^{K} Z_{ik}} \right) \frac{1}{C(\boldsymbol{\delta}^0)} \prod_{k=1}^{K} \pi_k^{\delta_k^0-1} \mathrm{d}\boldsymbol{\pi} \tag{61}$$

$$= \frac{C(\boldsymbol{\delta})}{C(\boldsymbol{\delta}^0)} \int_{\boldsymbol{\pi}} \mathrm{Dir}(\boldsymbol{\pi};\boldsymbol{\delta})\mathrm{d}\boldsymbol{\delta} \tag{62}$$

$$= \frac{C(\boldsymbol{\delta})}{C(\boldsymbol{\delta}^0)}, \tag{63}$$

where we used the abbreviation $C(\boldsymbol{\delta}) = \frac{\prod_{k=1}^{K}\Gamma(\delta_k)}{\Gamma(\sum_{l=1}^{K}\delta_k)}$ for $\boldsymbol{\delta} \in \mathbb{R}^K$. We take the negative logarithm of Eqn. (59) and eqn. (63) together with eqn. (55), this yields eqn. (12). □

For reference, we repeat the ICL_{ex} for the Bernoulli SBM introduced in [9]:

$$ICL_{ex}[\boldsymbol{Z},K] = \ln\left(\frac{\Gamma\left(\sum_{x=1}^{K}\delta_x\right)\prod_{x=1}^{K}\Gamma(\delta_x^0)}{\Gamma\left(\sum_{x=1}^{K}\delta_x^0\right)\prod_{x=1}^{K}\Gamma(\delta_x)} \right) + \sum_{x,y}^{K} \ln\left(\frac{B(\alpha_{xy},\beta_{xy})}{B(\alpha_{xy}^0,\beta_{xy}^0)} \right), \tag{64}$$

where $\boldsymbol{Z}$ is the hard cluster partition matrix, K the number of clusters, $\alpha_{xy}^0 = 1/2$, $\beta_{xy}^0 = 1/2$, $\delta_x^0 = 1/2$, α_{xy} is calculated according to eqn. 28, β_{xy} according to eqn. 29 and δ_x according to eqn. 30.

References

[1] T. A. Snijders and K. Nowicki. Estimation and prediction for stochastic block-models for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

[2] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5:109–137, 1983.

[3] J-J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statist. Comput.*, 18:173–183, 2008.

[4] M. Mariadassou, S. Robin, and C. Vacher. Uncovering latent structure in valued graphs: A variational approach. *Annals of Applied Statistics*, 4:715–742, 2010.

[5] J. M. Hofman and C. H. Wiggins. Bayesian approach to network modularity. *Phys. Rev. Lett.*, 100(258701), 2008.

[6] P. Latouche, E. Birmelé, and C. Ambroise. Variational bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, 12:93, 2012.

[7] R. P. Feynman. *Statistical Mechanics, A Set of Lectures*. W.A. Benjamin, Reading, MA, 1972.

[8] S. Abe and N. Suzuki. Scale-free network of earthquakes. *Europhys. Lett.*, 65(4):581, 2004.

[9] E. Côme and P. Latouche. Model selection and clustering in stochastic block models with the exact integrated complete data likelihood. *arXiv:1303.2962*, 2013.

[10] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review Letter E*, 69, 2004.

[11] V.D. Blondel, J.-L. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10:10008–10020, 2008.

[12] A. Channarond, J.J. Daudin, and S. Robin. Classification and estimation in the stochastic block model based on the empirical degrees. *arXiv:1110.6517v1 [math.ST]*, 2011.

[13] V. Miele et al. Technical documentation about estimation in the ermg model. http://stat.genopole.cnrs.fr/logiciels/mixnet/mixnet, 2007.

[14] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification. *J. Stat. Mech.*, P09008, 2005.

[15] A. McDaid, T. Murphy, and F.N.N. Hurley. Improved bayesian inference for the stochastic block model with application to large networks. *Computational Statistics and Data Analysis*, 60:12–31, 2013.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on PAMI*, 22:888–905, 8 2000.

[17] S. Abe and N. Suzuki. Complex-network description of seismicity. *Nonlin. Processes in Geophys.*, 13:145–150, 2006.

[18] S. Abe and N. Suzuki. Complex earthquake networks: Hierarchical organisation and assortative mixing. *Phys. Rev. E*, 74(026113), 2006.

[19] S. Abe and N. Suzuki. Aftershocks in modern perspectives: Complex earthquake network, aging, and non-markovianity. *arXiv:1202.4394v1 [physics.geo-ph]*, 2012.

[20] SCDEC. Southern california earthquake center caltech dataset, 2013. doi:10.7909/C3WD3xH1.

[21] K. Hutton, J. Woessner, and E. Handson. Earthquake monitoring in southern california for seventy years. *Bulletin of the Seismological Society of America*, 100:423–446, 2010.

[22] Y. Li and Z.-L. Zhang. Digraph laplacian and the degree of asymmetry. *Internet Mathematics*, 8(4):381–401, 2012.

[23] P. Wessel et al. GMT: The generic mapping tools, Version 4. http://gmt.soest.hawaii.edu, 2003.

[24] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 186(1007):453–461, September 1946.

[25] J. Kerman. Neutral non informative and informative conjugate beta and gamma prior distributions. *Electronic Journal of Statistics*, 5:1450–1470, 2011.

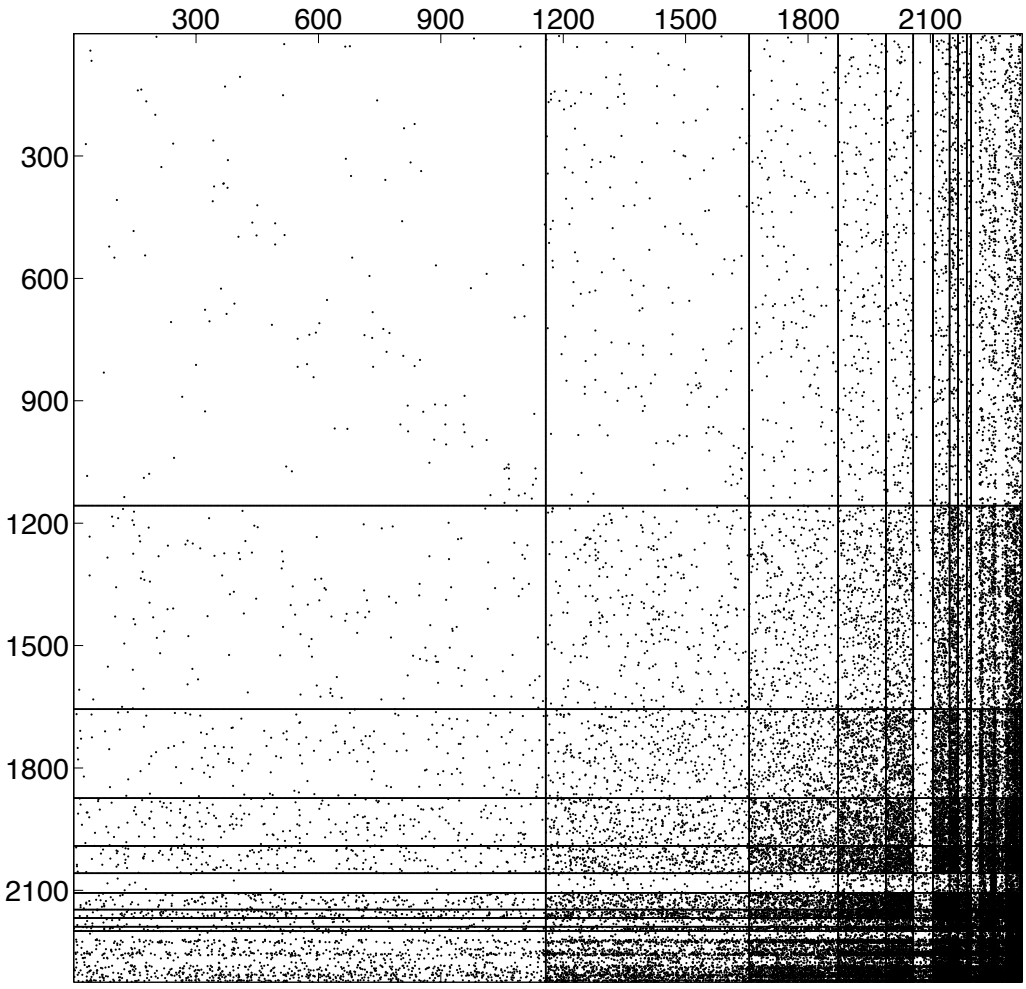

Figure 2: Dot–dot representation of the adjacency matrix of the Southern California Earthquake Network. The adjacency matrix is ordered according to the number of vertices per cluster. The clustering is the best result of the Blockloading algorithm for the Poisson SBM. The first 10 of the 46 clusters are separated with black lines.

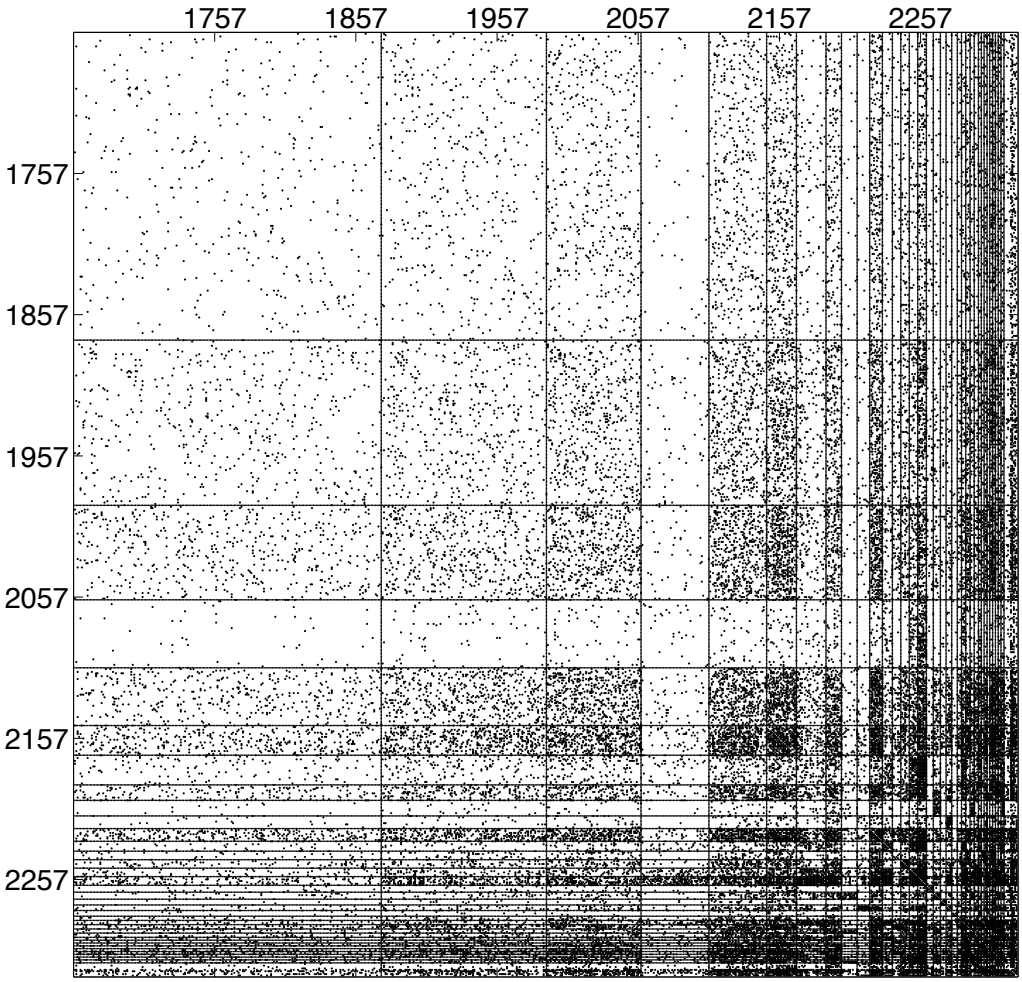

Figure 3: Dot–dot representation of an excerpt without the biggest two clusters of the adjacency matrix of the Southern California Earthquake Network. The adjacency matrix is ordered according to the number of vertices in each cluster in descending order. The first 35 clusters shown are separated with black lines. The smallest nine clusters are not separated by a line because there is only one vertex in each cluster.