EMPIRICAL BAYES ESTIMATION FOR RANDOM DOT PRODUCT GRAPH REPRESENTATION OF THE STOCHASTIC BLOCKMODEL

A thesis submitted in partial fulfillment of the requirements for the
Degree of
Doctor of Philosophy

SHAKIRA SUWAN
BSc (Hons)

under the supervision of
Dr Dominic Savio Lee
and
Professor Carey Priebe, Dr Carl Scarrott, and Associate Professor Donniell Fishkind

Department of Mathematics and Statistics
University of Canterbury
New Zealand

2015

[This page intentionally left blank]

ABSTRACT

Network models are increasingly used to model datasets that involve interacting units, particularly random graph models where the vertices represent individual entities and the edges represent the presence or absence of a specified interaction between these entities. Finding inherent communities in networks (i.e. partitioning vertices with a more similar interaction pattern into groups) is considered to be a fundamental task in network analysis, which aids in understanding the structural properties of real-world networks. Despite a large amount of research on this task since the emergence of graphical representation of relational data, this still remains a challenge. In particular, within the statistical community, the use of the stochastic blockmodel for this task is currently of immense interest.

Recent theoretical developments have shown that adjacency spectral embedding of graphs yields tractable distributional results. Specifically, a random dot product graph formulation of the stochastic blockmodel provides a mixture of multivariate Gaussians for the asymptotic distribution of the latent positions estimated by adjacency spectral embedding. The first part of this thesis seeks to employ this new theory to provide an empirical Bayes model for estimating block memberships of vertices in a stochastic blockmodel graph. Posterior inference is conducted using a Metropolis-within-Gibbs algorithm. Performance of the model is illustrated through Monte Carlo simulation studies and experimental results on a Wikipedia dataset. Results show performance gains over other alternative models that are considered.

Instead of a complete classification of vertices via community detection, one may wish to discover whether vertices possess an attribute of interest. Given that this attribute is observed for a few vertices, the goal is to find other vertices that possess that same attribute. As an example, if a few employees in a company are known to have committed fraud, how can we identify others who may be complicit? This is a special case of community detection, known as *vertex nomination*, which has recently grown rapidly as a research topic. The second part of this thesis extends the empirical Bayes model for vertex nomination based on information contained in the graph structure. This yields promising simulation results as well as real-data results from an Enron email dataset.

Recent studies have shown that information pertinent to vertex nomination exists not only in the graph structure but also in the edge attributes (Coppersmith and Priebe, 2012; Suwan et al., 2015). This motivates the third part of this thesis by further extending the model to exploit both graph structure and edge attributes for vertex nomination. Simulation studies confirm the benefit of doing so. However, the same benefit is not observed when the model is applied to the Enron email dataset; further investigations suggest that this is due to the data violating one of the model assumptions.

[This page intentionally left blank]

ACKNOWLEDGMENTS

First and foremost, I would like to thank God Almighty for giving me the patience and strength to complete this thesis.

I would then like to express my deepest gratitude to all my supervisors. This thesis would not have been achievable without the encouragement, patience and guidance I received from them, their efforts throughout this entire course has taught me a new set of skills which I never imagined I could possess. My biggest thanks go to Dr Dominic Lee for being the best supervisor, teacher and mentor I could ever ask for. Even after moving on from the world of academia, he still arranged regular meetings after his work hours to continue his support for me. I am very grateful for his precious time, effort, patience, support, and most importantly for believing in me. Without him I would not have courage and strength to acquire this highest level of qualification. A huge thank you to Professor Carey Priebe for his expertise and guidance throughout this process, in fact the ideas in this thesis mainly stem from the work of Professor Carey Priebe and his associates. I would like to thank his many associates who commented, shared their expertise, and contributed to making this thesis a worthwhile endeavor. I would also like to thank Dr Carl Scarrott for agreeing to be my supervisor, supporting, giving good advice from as early on as my undergraduate years, and encouraging me to pursue a PhD study when I doubted myself.

Many thanks to all the postgraduate colleagues and staff of the School of Mathematics and Statistics at the University of Canterbury for making my time as a PhD student very enjoyable. I am especially grateful to have had Rachel Binny as my office-mate. I can not imagine having a better colleague to work next to, thank you for being a good friend and supporting me through the many ups and downs in the last three years. And of course I cannot forget Paul Brouwers and Steve Gourdie, who were very friendly and approachable whenever I needed their assistance with my programming and computing troubles.

I need to also thank my mother, husband, children, in-laws, for their unconditional love, understanding, practical and emotional support, and for their confidence in me. For putting up with me when I was stressed out, and for being patient with me when I could not always give them the time they deserved.

Lastly, I am grateful to the University of Canterbury for supporting me financially through a scholarship grant to complete this thesis, without which I would never have been able to take this huge leap forward in my career.

[This page intentionally left blank]

Contents

List of Figures

List of Tables

NOTATION

General symbols and graph notation

$\mathbb{R}^d$	set of all real numbers in dimension d
G	graph set which consists of an order pair (V, E)
V	set of vertices of G
E	set of edges of G
$\lvert V \rvert = n$	number of points in set V
A	$n \times n$ adjacency matrix
$\mathbb{I}_{\{S\}}(x)$	indicator function: 1 if $x \in S$; 0 otherwise

Linear algebra

$\langle X, Y \rangle$	dot product of two vectors
$\|X\|_F$	Frobenius norm of a vector X
$X \perp Y$	orthogonality
$\mathrm{rank}(Z)$	size of the largest collection of linearly independent column of matrix Z

General statistics and probability

X	random variable
$\mathbb{P}(B)$	probability of event B
F or $F(x)$	cumulative distribution function
f or $f(x)$	probability density or mass function
$X \sim f$	X has density f
$X_1, \ldots, X_n \overset{iid}{\sim} F$	samples of size n is independent and identically distributed from F
$f(x, y)$	joint probability density or mass function
$f(x \mid y)$	probability density of x condition on y
$f(x; \theta^*)$	mass function or probability density for $X \sim f(x; \theta = \theta^*)$
$\mathcal{L}(\theta) := f(x_1^*, \ldots, x_n^*; \theta)$	likelihood function for parameter θ given data $x^*, \ldots, x_n^*$
$\Theta \in \mathbb{R}^d$	parameter space
$\theta \in \Theta$	scalar or vector in the parameter space
$\mathbb{E}(X)$	expectation of X
$\Sigma(X)$	covariance matrix
$\mathcal{N}_d(\mu, \sigma^2)$	density function of a multivariate normal distribution
$\xrightarrow{\mathcal{L}}$	convergence in distribution

ABBREVIATIONS

Acronyms

AEBSBM	attributed empirical Bayes stochastic blockmodel
ASGE	adjacency spectral graph embedding
BIC	Bayesian information criterion
DAG	directed acyclic graph
EB	empirical Bayes
EBSBM	empirical Bayes stochastic blockmodel
ERGM	exponential random graph model
GMM	Gaussian mixture model
LSM	latent space model
LPCM	latent position cluster model
MCMC	Markov chain Monte Carlo
M–H	Metropolis–Hastings
RDPG	random dot product graph

1 INTRODUCTION

In this thesis we develop a new empirical Bayes model that uses recent theoretical advances on adjacency spectral graph embedding techniques (Athreya et al., 2015), to perform community detection in networks including vertex nomination (a type of persons of interest identification problem). There are a plethora of real-world networks with inherent communities; however, efficiently identifying them is challenge. Thus, building models and algorithms to discover hidden community structures by exploiting information encapsulated in the graph topology is of great interest in various disciplines, including social science, biology, and neuroscience.

The motivation behind this research is detailed in Section 1.1. Section 1.2 gives a summary of previous developments in probabilistic modeling for the community detection problem using Bayesian analysis as well as spectral partitioning techniques. The objectives and contributions of this research are discussed in Sections 1.3 and 1.4, respectively. Section 1.5 outlines the structure of the thesis followed by the details of relevant publications and papers in Section 1.6.

1.1 Motivation

Analysis of network data is currently of burgeoning interest in various fields including social sciences, biology, physics, computer science, as well as statistics. A network often consists of entities (e.g. people, genes, or neurons) and their interactions (e.g. friendships, protein interactions, or synapses) which can be conceptualized as graphs.

A graph is a collection of vertices and edges connecting pairs of vertices. Graph representation of networks is applicable in various fields including social networks (vertices may represent people with edges indicating social interaction), citation networks (who cites whom), connectomics (brain connectivity networks; vertices may represent neurons with edges indicating axon-synapse-dendrite connections, or vertices may represent brain regions with edges indicating connectivity between regions), and many others.

Depending on the inferential objective of the analysis, the size of the dataset, as well as the nature of the data, various statistical approaches have been proposed to analyze network data starting with the simplest probabilistic random graph model of Erdös and Rényi, which exploited various properties of Bernoulli graphs and uniform graph models (Erdös and Rényi, 1959). However, these random graph models are not appropriate for modelling existing real-world networks due to the independent edges assumption and equal probability between pairs of connected vertices. To overcome these issues different models have been formulated in the literature (details follow in Section 2).

In many disciplines where data can be represented as graphs, identifying hidden community structures by exploiting the information encoded in the graph topology is often a primary concern. Community in this context implies groups of vertices with many edges connecting them within the same groups, and comparatively fewer edges connecting vertices of different groups, as depicted in Figure 1.1.

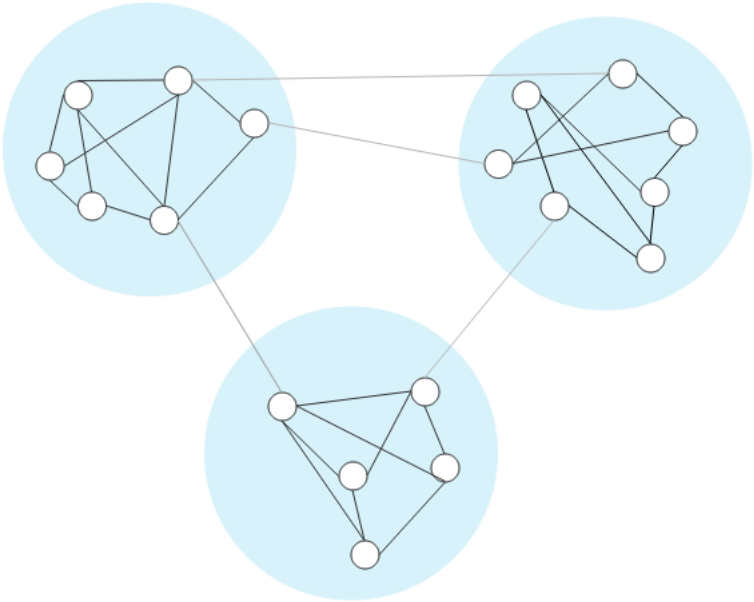


FIGURE 1.1: A small-scale network with a hidden community structure. There are three communities present here, illustrated by the highlighted blue color, which have more internal edges within groups but less edges between groups.

Hence, groups or communities of vertices often seemingly have similar attributes and/or roles within the graph. There are existing communities in various network systems; for instance, in the World Wide Web networks communities may correspond to groups of pages concerning similar or related topics (Flake et al., 2002), in protein-protein interaction networks they may represent groups of proteins which consist of similar function within cells (Rives and Galitski, 2003; Spirin and Mirny, 2003), in social networks groups may correspond with related individuals (Girvan and Newman, 2002), and others.

Community detection on graphs not only allows us to identify groups and objects of interest but also exposes the overall graph structure. This task is of significance in various application domains. As an example, consider the network of purchase connections between products of online merchants (eg. `www.ebay.com`) and customers. Discovering communities of customers with common interests and close geographical proximity allows merchants to improve their recommendation systems (Resnick and Varian, 1997) in order to better direct customers through their product list as well as increase business opportunities (Fortunato, 2010).

In addition, community detection can be conceived as a data mining analysis or clustering problem on graphs, which seeks to partition large sets of data into homogeneous groups or clusters. A well-known probabilistic model that addresses this task in network analysis is the stochastic blockmodel (SBM) (Holland et al., 1983). This model is an extension of the Erdös and Rényi random graph model and falls under the class of the latent position random graph model of Hoff et al. (2002). In this model, each of the n vertices is randomly assigned to pre–specified groups, and two vertices within the same latent group have the same probability of interactions with other vertices. Further, the presence of edges are conditionally independent given the block memberships of the vertex pairs. This type of model has been shown to be useful in detecting clusters within the network if the stochastic equivalence assumption holds. Many approaches have been developed to estimate block membership of vertices in an SBM graph; see Goldenberg et al. (2009) for comprehensive recent reviews.

In parallel to the SBM, Hoff et al. (2002) developed the latent position model for random graphs that provides a framework in which graph structure is parametrized by a latent position in $d-$dimensional Euclidean space associated with each vertex. The probability of an edge connecting two vertices is given by an appropriate function of the two corresponding latent positions. In particular, this thesis considers the special case of the latent position model known as the random dot product graph model (RDPG) which was introduced in Nickel (2006) and Young and Scheinerman (2007). In the RDPG, the probability of an edge connecting two vertices is given by the dot product of the corresponding latent positions. Recently, Sussman et al. (2012a) described a method for estimating the latent positions in an RDPG using a truncated eigen-decomposition of an adjacency matrix commonly known as adjacency spectral graph embedding (ASGE). This provides a technique to embed a graph as points in finite dimensional Euclidean space which allows the collection of statistical and machine learning methodologies to be utilized for graph inference. This new embedding procedure has inspired many researchers to further explore it in the context of various random graph models, with an emphasis on the outcome of subsequent inference.

For an RDPG, Athreya et al. (2015) presents distributional results of the residuals between the estimated and true latent positions equivalent to the central limit theorem in classical statistics. Specifically, they showed that the estimated latent positions (embeddings) via ASGE converge in distribution to a mixture of Gaussians. This crucial finding allows us to further analyze and accurately draw inference about the embeddings via standard multivariate methodology.

Despite the significant amount of literature on community detection in networks, this topic is still very challenging particularly with regard to recovering block membership assignments in the SBM graph. Furthermore, within the Bayesian framework the empirical Bayes approach has not been widely researched for this task. Hence, the first part of this thesis seeks to employ the distributional results of Athreya et al. (2015) to advance the performance of community finding in an SBM graph via empirical Bayes estimation.

Consider a network containing a subset of interesting vertices whose identities are not fully known, only a few of them are known; vertex nomination is a task which seeks to discover such interesting vertices. The meaning of "interesting" depends on the application context. As an example of application, suppose a small number of employees in a company have committed commercial fraud, but the identities of only a few of the fraudsters are known and we wish to use email communications between employees to identify one or more of the unknown fraudsters. Another example in law enforcement is to identify and prioritize child abuse offenders using the logging of peer-to-peer activities of individuals on child pornography networks; this does not require a warrant as there is a small percentage of individuals arrested for child pornography possession who are also child abusers. Moreover, it would also be of great importance for an affected country's national security bureau to use known terrorist profiles to identify other connected terrorists hidden within the community. This type of application can have a profound impact on the prevention of terrorist activity. It is apparent that vertex nomination is relevant, not only to the law enforcement and intelligence communities, but also in various social and business contexts; for instance the recommender systems (Resnick and Varian, 1997), the Netflix challenge (Bell et al., 2008) and detecting communities of interest (Cortes et al., 2001).

Vertex nomination is a special example of community detection, and is becoming a subject of significant study. Thus, a natural extension of our novel empirical Bayes model to perform vertex nomination is of theoretical and practical interest. This extension forms a second part of the thesis.

Using an attributed graph as a formalism to encode network data is also of great interest. Most real-world networks often inherently contain a rich set of attributes or characteristics attached to each vertex. For instance, in social networks profile information of individuals such as names, genders, or ages can be encoded as vertex–attributes. Similarly, we can have additional information about the relationship of the vertex pairs, such as communication topics and languages embedded as attributes associated to the edges. Attributed graphs are becoming increasingly prevalent in network modeling for representing a broad variety of data since exploiting such graphs can potentially be fruitful in various inferential goals. While there is a vast literature on random graphs as well as certain attributed graphs, not much is known about the random graphs with attributes (Grothendieck et al., 2010).

Recent studies have shown that information relevant to vertex nomination appears not only in the graph structure (context) but also in its attributes (content) (Coppersmith and Priebe, 2012; Coppersmith, 2014; Suwan et al., 2015). This suggests that utilizing both content and context information derived from an attributed graph can give superior vertex nomination performance than using one on its own. This motivates the last part of the thesis by further extending the empirical Bayes model to jointly exploit graph structure and edge attributes for vertex nomination.

1.2 Previous Research

Community detection is a crucial task in analyzing network data with immense significance across a broad spectrum of application domains such as biology, sociology, computer science, and other fields where systems can be represented as graphs. Searching for clusters/groups is algorithmically arduous since it is computationally intractable even on a small network to explore all possible clusterings. Thus, researchers have been developing various algorithms to alleviate these computational challenges by finding the "optimal" clusters (see Fortunato (2010) for comprehensive reviews).

To date, two main groups of methods have been identified in the literature to detect community structure in networks (Rohe et al., 2011). The first group is fitting schemes from heuristics or observations based on what the resemblance of network communities should be. For instance, a number of greedy algorithms such as spectral graph partitioning, the Girvan–Newman algorithm (Girvan and Newman, 2002), and hierarchical clustering (Newman, 2004), which essentially involve eigen decompositions. As these techniques are not concerned with a definitive probabilistic model it is beyond the scope of this thesis. Thus, we do not discuss them further here, apart from noting that they contribute to the formulation of subsequent statistical models. Another group is from various probabilistic model-based methods of a network with hidden communities such as the SBM and latent position random graph models.

An SBM has been studied and extended extensively in various contexts over the years with many inference objectives. One of the most important objectives is finding inherent community structures within the network. There are essentially four approaches with regard to the estimation of vertex block memberships. The first two methods being likelihood maximization (Bickel and Chen, 2009; Choi et al., 2012; Celisse et al., 2012; Bickel et al., 2013) and maximization of modularity (Newman and Girvan, 2004; Newman, 2006) which are both computationally challenging but not applicable in this study. The other two methods are spectral techniques and Bayesian methods. Relevant previous studies are discussed below.

While this thesis primarily focuses on correctly assigning vertices into their correct block under an SBM in a Bayesian perspective, it also relies heavily on an RDPG as well as theory and results within spectral graph embedding literature. Hoff et al. (2002) formulated latent space models for random graphs providing a framework in which a graph structure can be represented by latent positions tied to each vertex. The existence of an edge between a vertex pair is an independent Bernoulli trial conditioned on the latent positions. A special type of this model is a RDPG formulated by Nickel (2006) and Young and Scheinerman (2007) where the probability of an edge presence between two vertices is defined by a dot product of their corresponding latent positions.

Sussman et al. (2012a) used an embedding procedure motivated by the RDPG to estimate block membership of vertices in an SBM random graph. This embedding is analogous to the embeddings deployed in spectral clustering where the decomposition is operated on the normalized graph Laplacian; however it works directly on the adjacency matrix rather than a Laplacian matrix of the graph. Hence, the term "adjacency spectral graph embedding" (ASGE) is named for this procedure. They showed, via a nonparametric clustering algorithm, K–means, that clustering the embedded points can accurately assign the vertices into the correct blocks. They also adopted a unique way to define SBM as an RDPG representation which was later employed in a myriad of papers (Lyzinski et al., 2013; Athreya et al., 2015; Fishkind et al., 2013a). In this thesis, we will also employ this approach.

Shortly afterwards, Fishkind et al. (2013b) extended these aforementioned works of spectral partitioning results to the setting where the embedding dimension and the number of blocks are unknown. Athreya et al. (2015) extended the analysis of Sussman et al. (2012a) to show a distributional convergence, akin to the central limit theorem, for the residuals between the estimated and the true latent positions. They proved that for an RDPG, latent positions estimated using ASGE converge in distribution to a multivariate Gaussian mixture. This embedding technique is a subject for ongoing and future work in the context of various random graph models concentrating on the outcome of subsequent inference. These include Tang et al. (2013c) who extended this work to more general latent position models. Tang et al. (2013b) explored the out-of-sample extension for this graph embedding approach in the latent position graph model, while Tang et al. (2014) broadened the scope of this work to hypotheses testing on a pair of RDPGs. Lyzinski et al. (2014) used the aforementioned embedding technique together with existing state-of-the-art seeded graph matching procedures for large graphs, and Fishkind et al. (2013a) employed this work to address the vertex nomination task. Contrary to likelihood maximization and modularity methods these spectral techniques are not computationally expensive to implement.

Applying the Bayesian framework to network modeling has been previously explored extensively even outside of SBM dating back to Wong (1987) who extended the p_1 model of Holland and Leinhardt (1981) for directed graphs by employing the empirical Bayes procedure to estimate the model parameters, now known as the Bayesian p_1 model. Snijders and Nowicki (1997) used a Bayesian approach in estimating the block memberships from posterior predictive distribution for an undirected graph following an SBM for the case of $K = 2$ blocks via Gibbs sampling, and showed that the performance of the block structure estimation is better than maximum likelihood estimation. Soon after, Nowicki and Snijders (2001) extended the work of Snijders and Nowicki (1997) and developed a Bayesian probabilistic approach to a posteriori blockmodelling for directed graphs where the prior is a product of independent Dirichlet distributions with the posterior inference implemented via Gibbs sampling. The model assumed that *dyads* (pairs of vertices and their possible associated edges) in a network are conditionally independent, given the latent class membership of each vertex where vertices within a latent class are stochastically equivalent.

Gill and Swartz (2004) applied a full Bayesian analysis on directed graphs using a p_1 model that attempted to relax the dyadic independence assumption by using random-effects models. This is a generalization of the fixed-effects Bayesian p_1 model of Wong (1987). Airoldi et al. (2008) who developed the mixed membership SBM by using the empirical Bayes approach to estimate the model hyperparameters which have shown to perform well as the empirical Bayes method tends to direct the posterior distribution into the neighborhood of the hyperparameter space that is supported by the data. Their model differs from the existing SBM in the sense that they allowed each vertex to be in more than one cluster through a membership probability-like vector. The variational inference algorithm was employed for the posterior inference.

Rodríguez (2012) extended SBM to settings where the interaction of vertices can be observed more than once at different points in time to find any structural changes in the features that commonly appear in networks such as:

- homophily by attributes (the tendency of vertices with similar or completely opposite features to have a higher probability of presenting a link),
- differential attachment (some vertices tending to be more popular than others),
- transitivity (if vertex i and j are connected as well as j and k, then i and k are also connected),
- clustering (grouping of unlabeled data based on similarity)

as the network progresses. The model was built on generalized linear models which capture features of the networks with inference done within a Bayesian paradigm.

Additionally, Handcock et al. (2007) developed the latent position cluster model– a Bayesian methodology for clustering network data. This is an extension of Hoff et al. (2002), where all the key features of network data defined above are incorporated simultaneously - namely transitivity, clustering and homophily on attributes - which the existing SBMs struggled to represent. This work supposed that the latent positions are drawn from a mixture of multivariate normal distributions. However, finding the optimal number of underlying latent components in the mixture distribution is often a difficult task. An approximate form of the Bayesian information criterion (BIC) is used for this purpose in Handcock et al. (2007) where the computation of the maximum log likelihood was not achievable. Thus, Friel et al. (2013) recently proposed a more efficient alternative for this task by using conjugate prior distributions, allowing nearly all latent mixture parameters to be integrated out, and as such posterior inference can then be achieved conveniently via the Metropolis–within–Gibbs algorithm.

1.3 Thesis Objectives

The work in this thesis is divided into three parts, each of which is discussed in a separate chapter. The three key objectives of the research are:

- To develop an empirical Bayes model for estimating block memberships of vertices in an SBM random graph.
- To extend the empirical Bayes model to perform vertex nomination by exploiting a partially observed SBM graph.
- To further extend the empirical Bayes model for vertex nomination by exploiting a partially observed attributed SBM graph.

Here, a "*partially observed*" SBM graph is one for which the block membership is observed only for some (not all) of the vertices. An "*attributed*" graph is one whose edges contain additional information beyond their mere existence or non-existence.

1.4 Thesis Contributions

A primary contribution of this research is formulating a novel empirical Bayes model for estimating block memberships of vertices in an SBM graph and demonstrating its practical utility. To accomplish this, we use an alternative parametrization of an SBM as an RDPG model, wherein all vertices that belong to the same block share a common latent position. The proposed model is motivated by the Athreya et al. (2015) central limit theorem for RDPGs, which provides the asymptotic distribution of the latent positions estimated by ASGE as a mixture of Gaussians. Thus, we may consider the estimated latent positions of a K-block SBM to be independent and identically distributed from a (approximate) mixture of K-component multivariate Gaussians. The re-casting of the SBM as an RDPG and Athreya et al. (2015)'s results allow us to obtain an empirical prior for the unknown latent positions from the ASGE of the observed graph. Specifying an empirical prior on the unknown latent positions directly, rather than putting a prior on the block probability matrix, is one key contribution that distinguishes this work from other previous Bayesian approaches for estimating block memberships in an SBM graph.

Inference with the model is conducted using a Metropolis-within-Gibbs algorithm and performance is illustrated by three Monte Carlo simulation studies and one real data experiment. In all cases, the results show favorable performance relative to two benchmark models, and significant improvements over two other alternative models.

Apart from community detection in graphs, this research also addresses vertex nomination. Another main contribution of this research is to develop a model for vertex nomination by extending the previous empirical Bayes model. This proceeds by treating vertex nomination as a two–block SBM with one of the two blocks containing the interesting vertices. Given that the block membership of a few interesting vertices is observed, the resulting model utilizes connectivity structure of the graph to perform vertex nomination. This requires a new likelihood function that incorporates the information about the observed interesting vertices, in conjunction with the previous prior specifications. The efficacy of the model is demonstrated by two simulation studies and a real-data application involving a subset of the Enron email corpus. The nomination performance of our model is compared against those of Coppersmith and Priebe (2012) and Suwan et al. (2015). It is worth noting that their models utilize additional edge information not used in our model. Despite this, the results show that the performance of our model is comparable to theirs.

Yet another contribution is further extending our vertex nomination model to utilize both graph structure and edge-attributes. This involves extending the likelihood function to capture the additional information about the edge attributes, as well as appropriate extension of the prior specifications. These extensions require the implementation of a new Metropolis-within-Gibbs algorithm for posterior inference. In a simulation study, this extended vertex nomination model provides a substantial nomination performance gain over both the previous model and the method of Suwan et al. (2015). The improvement over the previous model reinforces the recent findings (Coppersmith and Priebe, 2012; Suwan et al., 2015) that vertex nomination performance will benefit by leveraging both graph structure and edge attributes. However, when applied to the same Enron email dataset used previously, no performance gain is evident. A second simulation study conducted to explain this suggests that this is due to the data violating a model assumption.

Although community detection in graphs has been studied since the onset of graphical representation of relational data, vertex nomination is relatively new with much room for further exploration. Overall, this thesis introduces an empirical Bayes model for community detection and extends it into two other models for vertex nomination. Simulation studies and real-data applications suggest that the proposed models are a worthwhile contribution to the statistical network modeling literature.

1.5 Thesis Structure

Chapter 2 (Background Material) further provides the literature review of relevant materials, a summary of statistical concepts, and useful results within statistical fields required in Chapters 3, 4, and 5. Particularly, reviews concentrate on the pertinent random graph models, Bayesian analysis, and adjacency spectral embedding. Note that background material and previous works for Chapters 4 and 5 pertaining to vertex nomination and an attributed graph are addressed within their respective chapters.

Chapter 3 (Empirical Bayes Estimation for the Stochastic Blockmodel) is concerned with a new approach that uses Bayesian inference to detect communities in networks. The approach relies primarily on an RDPG and an SBM. This involves parametrizing SBM as RDPG and adopting adjacency spectral embedding theory to construct an empirical prior on latent positions. We introduce two benchmark models (named *Exact* and *Gold*) as well as an alternative *flat* model for the purpose of subsequent model comparison. Inference with these models is conducted using a Metropolis–within–Gibbs algorithm. The performance is illustrated by a Monte Carlo simulation study and by application to a Wikipedia network.

Chapter 4 (Vertex Nomination via Empirical Bayes Estimation) considers extending the concepts of Chapter 3 to perform vertex nomination on a partially observed non-attributed SBM graph. To facilitate this, a new likelihood model incorporating the additional information from the few interesting vertices is formulated. As in Chapter 3, a graph realized from the SBM is exploited, but a key difference is that the block memberships of a few vertices in the graph are assumed observed whilst the rest of the vertices' block memberships are assumed unobserved. The vertex nomination problem is introduced along with its relevant literature review. Simulation studies are used to compare our model with the existing works of Coppersmith and Priebe (2012) and Suwan et al. (2015). The model is also applied to the famous Enron email dataset. Since our exploitation task involves finding one or more of the interesting vertices rather than a complete classification, performance measures including minimum reciprocal rank (MRR), mean average precision (MAP), and precision at rank k are considered.

Chapter 5 (Extending Vertex Nomination via Empirical Bayes Estimation to Attributed Graphs) advances our vertex nomination model in Chapter 4 by exploiting attributed graphs. This is achieved by constructing a new likelihood model encapsulating the additional information from the edge attributes together with the previous prior specifications for the model parameters. Inference about the unknown vertices can then be obtained from the resulting posterior distribution, allowing the construction of the nomination list of vertices with those at the top of the list having the highest posterior probability that are most likely to be interesting. In the same manner as Chapter 4, inference of the model is conducted using a Metropolis–within–Gibbs sampler, and a Monte Carlo simulation study is used to illustrate the model's performance and experimental results on the Enron graph. These results are then compared with Suwan et al. (2015) and the method proposed in Chapter 4.

Chapter 6 (Concluding Remarks) summarizes the thesis and addresses future research areas with reference to the results in this thesis.

1.6 Thesis Publications

Paper published

Suwan, Shakira, Dominic S. Lee, and Carey E. Priebe. "Bayesian vertex nomination using content and context." *Wiley Interdisciplinary Reviews: Computational Statistics 7(6)*, 400-416.

Paper under review

Suwan, Shakira, Dominic S. Lee, Runze Tang, Daniel L. Sussman, Minh Tang, and Carey E. Priebe. "Empirical Bayes estimation for the stochastic blockmodel." *arXiv preprint arXiv:1405.6070 (2014).*

Paper in preparation

Suwan, Shakira, Dominic S. Lee, Carey E. Priebe, Carl Scarrott, Runze Tang, and Minh Tang. "Vertex nomination via empirical Bayes estimation."

Conference presentations

Suwan, Shakira, Dominic S. Lee, Carey E. Priebe, Carl Scarrott, Runze Tang, Daniel L. Sussman, and Minh Tang. "Vertex nomination via empirical Bayes estimation." *New Zealand Statistical Association Conference*, 24 - 27 November 2013, Hamilton, New Zealand.

Suwan, Shakira, Dominic S. Lee, Carey E. Priebe, Carl Scarrott, Runze Tang, Daniel L. Sussman, and Minh Tang. "Vertex nomination via empirical Bayes estimation." *Bayes on the Beach Conference*, 10 - 12 November 2014, Gold Coast, Australia.

2 | BACKGROUND MATERIAL

The novel methodological developments in this thesis extend various random graph models as well as spectral embedding techniques with all inferences carried out in a Bayesian context. This chapter reviews the pertinent literature and introduces key models and results that will be used in the thesis. Specifically, Sections 2.1 and 2.2 address various types of networks, and introduce notations and terminologies used in this thesis respectively. Section 2.3 reviews and describes pertinent random graph models, followed by spectral graph embedding theory in Section 2.4. Descriptions of the Gaussian mixture model as well as Bayesian analysis are presented in Sections 2.5 and 2.6.

2.1 Types of Networks

In general, a collection of vertices and edges connecting pairs of vertices is the most basic type of network. Networks may be more complex where there exists more than one type of vertices or edges with various properties attached. As an example, in social networks, vertices may represent people, locations, ages, different nationalities, or other such factors. Edges may indicate friendships, antagonism, or geographical proximity. Moreover, these edges can also be enriched by carrying weights, indicating, for instance, how close their friendships are. An edge can either be directed if it points to only one direction, or undirected if it points in both directions. For example, a network representing email messages of individuals where each message only goes to one person unreplied, would be composed of directed edges. Unsurprisingly, graphs containing directed edges are known as directed graphs or *digraphs* for short. There are two types of directed graphs, cyclic (self-loops), or acyclic (no self-loops). Further, edges that connect more than two vertices together are called *hyper–edges* and graphs containing such edges are similarly known as *hyper–graphs*.

2.2 Basic Notations and Terminologies

In this thesis, a graph or network, G, consists of an order pair, (V, E), which can be equivalently expressed as $G \equiv G(V, E)$, where V denotes a set of vertices, E denotes a set of edges, and $n = |V|$, $e = |E|$. In the simplest setting, network data is often represented as an $n \times n$ adjacency matrix, $A \in \{0, 1\}^{n \times n}$, of the binary relations, A_{ij}, between vertices, i and j. If an edge is present $A_{ij} = 1$ or 0 if not present. G is also frequently defined as a collection of the vertices and the corresponding adjacency matrix, A, $G \equiv G(V, A)$. The adjacency matrix of directed edges is asymmetric (i.e. $A_{ij} \neq A_{ji}$), and for undirected edges is symmetric (i.e. $A_{ij} = A_{ji}$) for all $i \neq j$. The presence of a directed edge between a pair of vertices, (i, j), is written as $i \rightarrow j$, and for an undirected edge as $i \sim j$.

2.3 Random Graph Models

Network data in general contains vertices which are pairwise related implying a violation of the classical independent assumption applied in traditional applications. This interdependent structure motivates the development of new approaches for exploiting network data; an ongoing active area of research. A plethora of different statistical approaches have been proposed in recent years to analyze network data depending on several factors such as the inferential objective of the analysis, the size of the dataset, as well as the nature of the data (Ghosh et al., 2010). These statistical network models often involve a form of graphical representation and focus on certain local and global network statistics.

We can divide statistical network models into two major categories: static models and dynamic models. Static models are concerned with analyzing the observed edge set from a single snapshot of the network, while dynamic models primarily concentrate on tools that address changes in the network over time. This thesis focuses specifically on static network models. Exploration of the prominent static network models and their associations are briefly presented in this section. A more detailed review on statistical models for networks has been compiled by Goldenberg et al. (2009) and Fortunato (2010).

Figure 2.1 displays a visual diagram of the connections between the prominent random graph models relevant to this thesis, which will be discussed below. An arrow from A to B can either signify that model B is a generalization of model A, or the development of model A motivated the formulation of model B. For example, p_1 model can be viewed as a generalization of the

Erdös-Rényi model (Section 2.3.2), whereas SBMs give rise to various model formulations such as degree–corrected SBM, weighted SBM, etc (Section 2.3.4).

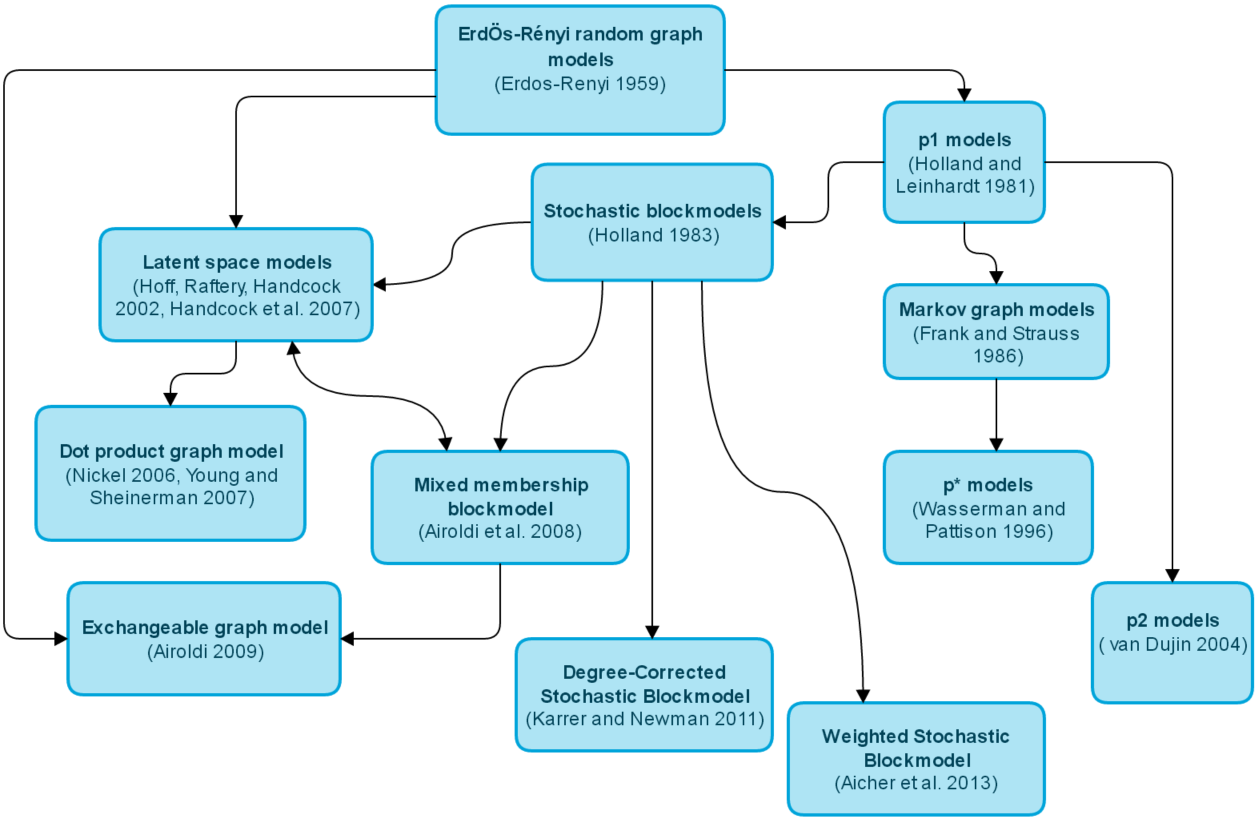


FIGURE 2.1: A summary of probabilistic graphical models addressed in Section 2.3.

2.3.1 Erdös-Rényi Model

One of the oldest and most basic probabilistic models for network data is the Erdös-Rényi model (Erdös and Rényi, 1959). In fact, the field of random graph theory originated from research on the Erdös-Rényi model (Goldenberg et al., 2009). This model describes an undirected graph for n vertices and a fixed number of edges, e, chosen at random from the $\binom{n}{2}$ possible edges in the graph. In addition, this model assumes that the presence or absence of edges between vertices are independent and identically distributed with the probability p if an edge is present and $1-p$ if otherwise. In terms of the $n \times n$ binary adjacency matrix, A, the likelihood function of the model is given by

$$f(A \mid p) = \prod_{i \neq j} p^{A_{ij}} (1-p)^{1-A_{ij}},$$

where for directed graphs the product encompasses all pairs, $i \neq j$, and for undirected graphs, $i < j$.

For modeling existing real-world network where the relationships between observations are complex and often display dependency, this random graph model is not appropriate due to the independent edges and equal probability assumptions (Salter-Townshend et al., 2012). Thus, it is used mainly as a null model where no structure is present.

Additionally, there remains a need to formulate formal tools in order to determine the performance of the model fit for a given observed graph, and decide on which types of generalized graph models are more suitable. This brought about two prominent lines of research, the first with a focus on feature model prediction and verification of those features in the observed networks (Goldenberg et al., 2009). These objectives are generally studied within the field of computer science and statistical physics which are outside the scope of this thesis. The second direction has been devoted to exploring formal statistical properties in relation to estimating parameters of network models, for example the p_1, p_2 and exponential random graph models described below.

2.3.2 p_1 and p_2 Models

The p_1 and p_2 models are types of logistic regression models. The p_1 model for directed graphs, introduced by Holland and Leinhardt (1981), is an extension of the Erdös-Rényi model with a slight increase in complexity. This model incorporates parameters for the number of relations, individual tendencies to give or receive relations, and parameters for the propensity of relations (some actors are likely to be more popular than others) to be reciprocal. All are based on four possible types of relationships between vertex i and j below.

- No link between i and j
- i links to j only
- j links to i only
- i links to j and j links to i

Let $\mathbb{P}_{ij}(0,0)$ be the probability of no edge between a vertex pair, i and j, $\mathbb{P}_{ij}(1,0)$ be the probability of i links to j where 1 denotes the outgoing vertex of the edge, $\mathbb{P}_{ij}(1,1)$ be the probability of i links to j and j links to i.

Thus, in the p_1 model, these probabilities can be expressed as

$$\begin{aligned}
\log\mathbb{P}_{ij}(0,0) &= \psi_{ij},\\
\log\mathbb{P}_{ij}(1,0) &= \psi_{ij} + \alpha_i + \beta_j + \theta,\\
\log\mathbb{P}_{ij}(0,1) &= \psi_{ij} + \alpha_j + \beta_i + \theta,\\
\log\mathbb{P}_{ij}(1,1) &= \psi_{ij} + \alpha_i + \beta_j + \alpha_j + \beta_i + 2\theta + \rho_{ij},
\end{aligned}$$

where ψ_{ij} is a normalizing constant to ensure that the probabilities sum up to 1 for each vertex pair, (i,j), α_i is the effect of an outgoing edge from vertex i and commonly referred to as expansiveness, β_j is the effect of an incoming edge into vertex j or popularity, ρ_{ij} is the reciprocation effect. Representing the p_1 model in this manner engenders a non-identifiability issue of the reciprocation parameters (see (Holland and Leinhardt, 1981) for more details).

Four special cases of the p_1 model that are identifiable and of particular interest include:

1. $\alpha_i = 0$, $\beta_j = 0$, and $\rho_{ij} = 0$. This p_1 model essentially reverts to an Erdös-Rényi model for directed graphs.

2. $\rho_{ij} = 0$. This is the case where there is *no reciprocal effect*, the model only captures the degree distributions into and out of vertices.

3. $\rho_{ij} = \rho$. This version of the p_1 model has *constant reciprocation* , which Holland and Leinhardt (1981) explored thoroughly via maximum likelihood estimation.

4. $\rho_{ij} = \rho + \rho_i + \rho_j$. It is an *edge–dependent reciprocation* version of the p_1 model, pursued by Fienberg and Wasserman (1981).

The likelihood function of the p_1 model falls under the exponential family form. For instance, in the case of the constant reciprocation, this comprises four main parameters capturing four types of connections outlined above, giving

$$f(A=a) \propto \exp\left(\theta\sum_{i,j} a_{ij} + \sum_i \alpha_i \sum_j a_{ij} + \sum_j \beta_j \sum_i y_{ij} + \rho m\right),$$

where a is the observed adjacency matrix, $m = \sum_{i<j} a_{ij}a_{ji}$ is the number of mutual links, and θ is a network-wide base rate for edge probability. Notably, this model can be parameterized as a generalized linear model (GLM) (see (Duijn et al., 2004) for a full expression).

The complications with these models include the lack of standard asymptotics to aid with the goodness-of-fit procedure formulation for the model and lack of consistent results for the maximum likelihood estimates. This stems from an increase in the number of α_i and β_j as the number of vertices increase in a linear manner. Moreover, these models also assume that the probability of an edge between two vertices is independent of the presence or absence of edges connecting any other pair of vertices which leads to an inability to capture common features of networks that involve more than two vertices (Hoff et al., 2002).

The p_1 model treats expansiveness, α_i, and popularity, β_j, as fixed effects (a set of unobserved constant quantities) corresponding to unique vertices in the network. However, it is more sensible to presume that a set of these effects is distributed according to some underlying distribution in which the estimation of its parameters are then computed. This motivated Duijn et al. (2004) to formulate the p_2 model, where the aforementioned parameters are treated as random effects (a set of unobserved variable quantities) that are drawn from some distributions. This can easily be extended from any of the multivariate variations on p_1 and yield a multi–level (hierarchical) model with fixed and random effects mixtures (for instance, see (Zijlstra et al., 2006)). Hence, the p_2 model is an extension of the p_1 model and is essentially a generalized linear mixed model. Bayesian inference on these models has been explored in many studies. A key difference between Bayesian extensions of the p_1 and the p_2 model is that the unknown constants in the model will be treated as random effects in the Bayesian method, yielding the models with more levels in the multilevel hierarchy (Wang and Wong, 1987; Gill and Swartz, 2004).

2.3.3 Exponential Random Graph Models

ERGMs are a family of models that attempt to tackle the issues resulting from the assumption of dyadic independence in the p_1 and p_2 models. They are also known as p^* models– essentially a generalized linear model for network data. These models are extended from the Markov random graphs of Frank and Strauss (1986) which assume that the presence or absence of an edge between two vertices is conditionally dependent on the edges that share one of the two vertices.

The probability distribution of undirected Markov graphs is given by

$$F(A = a) = \exp\left(\sum_{k=1}^{n-1} \theta_k S_k(a) + \tau T(a) + \phi(\theta, \tau)\right) \qquad a \in A,$$

where $\theta := \{\theta_k\}$ and τ are parameters of the model, $\phi(\theta, \tau)$ is the normalizing constant, and the statistics, S_k and T, are counts of specific structures such as edges, triangles, and k-stars (a connected graph with no cycles containing exactly one internal vertex with k leaves):

- number of edges: $S_1(a) = \sum_{1 \leqslant i \leqslant j \leqslant n} a_{ij}$,
- number of triangles: $T(a) = \sum_{1 \leqslant i \leqslant j \leqslant h \leqslant n} a_{ij} a_{ih} a_{jh}$
- number of k-stars: $(k \geqslant 2)$ $S_k(a) = \sum_{1 \leqslant n} \binom{a_{i+}}{k}$.

Wasserman and Pattison (1996) generalized the Markov graphs into ERGM where the statistics, S_k and T, are substituted by arbitrary statistics, U. Likelihood functions for ERGMs are of the form

$$f(A = a) = \exp\left(\theta^{\mathsf{T}} u(a) - \phi(\theta)\right),$$

where the counts of graph structures are denoted by the statistics $u(a)$.

ERGMs are advantageous in the sense that they are able to represent a variety of structural tendencies, for instance transitivity, propensities for homophily, mutuality, by a different choice of model parameters (sufficient statistics).

Nonetheless, it is generally problematic to obtain the normalizing constant, ϕ, under these models since the summation of all possible networks with the observed sufficient statistics, $u(a)$ is required. Thus, fitting ERGMs often involves estimating the parameters for each of the network statistic terms as well as the normalizing constant for the underlying model. There are various methods available for ERGM model fitting without summing over all possible networks, such as Monte Carlo maximum likelihood estimation(MCMLE) (van Duijn et al., 2009), Markov chain Monte Carlo (MCMC) (Caimo and Friel, 2011), and maximum pseudo–likelihood estimation (Strauss and Ikeda, 1990).

In various inferential methods, model degeneracy may be a problem. This is a case where only a few networks contain probability that is observable given the model. If this problem arises then the estimates of the parameters may not converge, resulting in misleading estimates when using, for instance, maximum pseudo–likelihood estimators. In such cases, the model is not properly specified and no estimation technique can be employed to overcome this. The reader is referred to Tjelmeland and Besag (2001), Snijders (2002), and Robins et al. (2007b) for further discussions on degeneracy issues in network statistical models and various new specifications of ERGMs to address this problem.

Recently, the cluster p^* model was also implemented by Steinley et al. (2011) who combined blockmodeling and p^* models. This is in contrast with traditional blockmodeling, in that the groupings are not based on the denseness of blocks but rather on the functional difference in terms of graph structure. In addition, the authors claimed that the cluster p^* approach avoided the notorious degeneracy problem by allowing local estimation of network groups rather than simultaneously estimating the entire network (Steinley and Wasserman, 2011). Another extension of p^* models was recently proposed by Ouzienko and Guo (2011) to predict both actor attributes and links in temporal networks. These were achieved by implementing two conditional predictors to simultaneously infer actor attributes and links.

2.3.4 Stochastic Blockmodels

Prior to stochastic blockmodelling, a technique known as blockmodelling was used to find an optimal partition of vertices in a network then classify them into clusters or blocks. Each block consists of vertices with similar properties or attributes in the network. A detailed discussion of non-statistical blockmodelling is beyond the scope of this thesis; see Carrington et al. (2005, Chapter 5) for overviews of blockmodelling techniques.

The first work to incorporate the ideas of blockmodelling into a probabilistic random graph framework by Holland et al. (1983), is considered to be the birth of the stochastic blockmodel. It is part of the general class of random graph models and has been researched extensively in the fields of computer science and the social sciences. The model is in fact an extension of the p_1 model but includes parameters describing differential rates of within-group and between-group relations in situations where vertices are assigned to pre-specified groups.

This model is popular for detecting community structure in unweighted networks. In its basic form, the stochastic block distribution, $\mathrm{SBM}(K, \rho, B)$, supposes that each of the n vertices, labeled $1, 2, \ldots, n$, is randomly assigned to one of K blocks by a random block membership function, $\tau : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, K\}$. The probability of an edge between two vertices depends only on their respective block memberships, while the presence of edges are conditionally independent given block memberships. By letting τ_i denote the block to which vertex i is assigned, a $K \times K$ matrix B is defined as the block probability matrix such that the entry, B_{τ_i,τ_j}, is the probability of an edge between vertices, i and j, i.e. $B \in [0,1]^{K \times K}$. The block proportions are represented by a K-dimensional vector, ρ, satisfying $\sum_{k=1}^{K} \rho_k = 1$. In other words, Wasserman and Anderson (1987) defined two vertices to be *stochastically equivalent* if they are assigned to the same block. The conditional probability function for the SBM is thus given by

$$\begin{aligned} f(A \mid \tau) &= \prod_{i \neq j} f\left(A_{ij} \mid \tau_i, \tau_j\right) \\ &= \prod_{i \neq j} B_{\tau_i,\tau_j}^{A_{ij}} (1 - B_{\tau_i,\tau_j})^{1-A_{ij}}. \end{aligned} \tag{2.1}$$

Hence, conditional on τ, the entry $A_{ij} | \tau_i, \tau_j \overset{ind}{\sim} \text{Bern}\left(B_{\tau_i,\tau_j}\right)$.

Notably, this model can generate a vast range of distinct network structures depending on the choice of the block probability matrix, B. For instance, with a diagonal matrix, B, the block structure will display groups with only edges present within groups. On the other hand, adding small off-diagonal elements, would produce community structures composing of groups with more links present internally and less between blocks. This particular setting is commonly referred to as an *affinity* SBM; a setting we explore in the remaining chapters. Other choices of B can produce hierarchical, core–periphery structures (the 'core' mostly having edges within itself, while the 'periphery' is predominantly linking to the 'core'), and many others (Clauset et al., 2008; Park et al., 2010).

In addition to this adaptability of the underlying graphical representation, the probabilistic structure of SBM also provides a way of assessing the uncertainty of block membership. This feature has brought about theoretical guarantees, such as the identifiability and consistency of latent block models (Allman et al., 2011; Sussman et al., 2012b, 2014; Tang et al., 2014) and the consistency of the SBM estimators (Bickel and Chen, 2009; Sussman et al., 2012a; Fishkind et al., 2013b).

Wang and Wong (1987) were one of the first to explore SBM by simply extending the p_1 model for the directed graphs of Holland and Leinhardt (1981) to account for the block structure using an extra block parameter where maximum likelihood was adopted to estimate the model's parameters. Anderson et al. (1992) provided a general definition of an SBM and considered several techniques for constructing such models for an observed network.

A primary component of an SBM is the function, τ, which maps vertices to blocks. There are two main distinctions in relation to estimating the number of blocks. When the ideas of block-modeling and a statistical methodology were initially put together, a priori blockmodelling, one of the distinctions, was a focus of attention as it assumes that the blocks are known through some exogenous attribute information on the vertices such as age, income, and gender (Wasserman, 1994). The other more challenging approach relies on the relational data and standard clustering techniques to obtain the number of blocks and is commonly referred to as a posteriori blockmodelling. In the remaining chapters of this thesis, we assume that the number of blocks is known a priori.

Wasserman and Anderson (1987) and Anderson et al. (1992) analyzed a posteriori strategy within the p_1 family by means of the p_1 model fitting to digraph data then partitioning the vertices with similar maximum likelihood estimates of expansiveness, α, and popularity, β, parameters together and lastly fitting the pair-dependent SBM. Similarly Snijders and Nowicki (1997) studied a posteriori blockmodelling for an undirected graph with two blocks via various statistical estimation and prediction methods such as the profile likelihood, the EM algorithm, the conditional predictive likelihood, and Gibbs sampling. Under mild conditions, they demonstrated that it is asymptotically possible to correctly recover the block memberships of vertices with probability approaching one. Based on their investigation, the Bayesian method with Gibbs sampling was relatively the most viable approach on graphs with the number of vertices larger than 15. Later Nowicki and Snijders (2001) generalized the Bayesian method of Snijders and Nowicki (1997) to examine a posteriori blockmodelling for digraphs which introduces a prior Dirichlet distribution for the model parameters. Inference for the model was conducted using a Gibbs algorithm. Other recent attempts to extend SBMs include Airoldi et al. (2008), who used a variational approximation to fit a mixed-membership SBM, Rodríguez (2012) who extended SBMs to settings where the interaction of vertices can be observed more than once with the goal of identifying structural changes in the features of the model network as the network progresses, and Latouche et al. (2012) who developed a variational Bayesian approach for complexity control in SBMs.

Given an SBM graph, estimating the block memberships of vertices is often an important task. Many approaches have been developed for the estimation of vertex block memberships, including likelihood maximization (Bickel and Chen, 2009; Choi et al., 2012), maximization of modularity, spectral techniques (Rohe et al., 2011; Sussman et al., 2012a; Fishkind et al., 2013a), and Bayesian methods (Snijders and Nowicki, 1997; Nowicki and Snijders, 2001; Handcock et al., 2007).

In many real-world networks, the existence of “hubs” or high-degree vertices at the center of communities are ubiquitous, thus modeling using the standard SBM may incur serious limitations as it presumes that all vertices within the same community are stochastically equivalent or have the same expected degree (Zhao et al., 2012). To tackle this issue, Karrer and Newman (2011) proposed the degree-corrected SBM, an extension of the SBM which allows for heterogeneous degrees. Moreover, a recent generalization of an SBM is the weighted SBM of Aicher et al. (2014) where edges can have weight attached to them. This extra information can give insight into the hidden community structures in edge-weighted networks without having to discard edge-weights or place thresholds prior to analysis; a procedure in which a basic SBM is often required. Weights on edges may sometimes refer to attributes on the edges, where a scalar weight signifies the connection's strength. Chapter 5 will exploit attributed graphs where each edge has a categorical variable attached to it rather than weight for the vertex nomination task.

2.3.5 Latent Space Models

The many restrictions in the previous models (Sections 2.3.2 and 2.3.3) placed more emphasis on the global rather than the local structure, resulting in model degeneracy and instability issues (Hoff et al., 2002). These complications inspired Hoff et al. (2002) to develop alternative models, namely the distance and the projection model, classified under the latent space model (LSM) for random graphs (Hoff et al., 2002).

This model presents a framework in which each vertex, $i \in V$, is associated with a latent position, X_i, in a d-dimensional Euclidean latent space, $X_i \in \mathcal{X} \subset \mathbb{R}^d$. Given the latent positions, the presence of each edge between vertex i and j, A_{ij}, is an independent Bernoulli trial. The edge presence probability is determined by the link function (or kernel), $\kappa : \mathcal{X}^2 \mapsto [0, 1]$, a symmetric function of the two latent positions, which returns the probability of these latent positions being connected in the resulting graph (Athreya et al., 2015). In general, as the distance between two vertices becomes smaller in these models their probability of having a tie becomes higher (Salter-Townshend et al., 2012). In the context of social networking, this can be interpreted as relationships of individuals formed depending on common characteristics, for instance proximity, or shared interests.

A prominent feature of these models is a natural capturing of the structural tendencies in network relations such as transitivity, homophily by attributes, and reciprocity (the probability that two vertices in a directed graph point to each other) in the case where additional edge covariate information is available.

The conditional probability model for the adjacency matrix, A, is

$$f(A \mid X, Z, \Theta) = \prod_{i \neq j} f(A_{ij} \mid x_i, x_j, Z_{ij}, \Theta),$$

where $X \subset \mathbb{R}^d$ are the latent positions of vertices, Z are covariates, and Θ are parameters.

As mentioned above, Hoff et al. (2002) developed two LSMs− the distance model and the projection model. The former uses the Euclidean distance in the social space as the link function. However, any distance, $d_{ij} = d(X_i, X_j)$, complying with the triangle inequality, $d_{ij} \leqslant d_{ik} + d_{kj}$, for all $\{i, k, j\}$ may be considered.

The distance model uses the logistic regression model to parametrize $f(A \mid X, Z, \Theta)$ which can be expressed as

$$\log \frac{f(A_{ij} = 1)}{1 - f(A_{ij} = 1)} = \alpha + \beta^{\mathsf{T}} Z_{ij} - |X_i - X_j| \equiv \eta_{ij},$$

where Z_{ij} is a pair specific covariates and $\Theta = (\alpha, \beta)$ is a set of parameters to be estimated.

This gives the corresponding log likelihood as

$$\log f(A \mid \eta) = \sum_{n \neq m} \left(\eta_{ij} \times A_{ij} - \log(1 + e^{\eta_{ij}})\right).$$

Due to the model's simple interpretation as the connection relies directly on the distance between the vertex pair in the social space, this is the most preferred model with the distance being the Euclidean distance. In this model, the graph is assumed to be inherently symmetric because of the reciprocity feature. That is if $a_{ij} = 1$ then there is a high chance that $a_{ji} = 1$. Thus, the distance model is particularly good for both directed and undirected graphs that show strong reciprocity.

Similarly, the projection model also uses the logistic regression model as a parametrization of the probability of an edge between vertex i and j which is written as

$$\eta_{ij} = \alpha + \beta^{\mathsf{T}} Z_{ij} - \frac{|X_i^{\mathsf{T}} X_j|}{|X_j|}.$$

The projection model is more suitable for networks that are strongly asymmetric. This is because of the assumption that the edge presence probability of two vertices is defined as the angle in the bilinear latent space, therefore, the probability of an edge presence is high if the angle is small and vice versa. The inference for social space was carried out within both the Bayesian and maximum likelihood frameworks.

Shortly after, Handcock et al. (2007) extended the work of Hoff et al. (2002) by formulating the latent position cluster model (LPCM) which incorporates all the key features of network data simultaneously, namely transitivity, clustering and homophily on attributes that existing SBMs struggle to represent. They developed two methods for estimating the latent positions and the model parameters. Firstly, the two-stage maximum likelihood method initially maps the vertices in the latent space and subsequently uses a finite Gaussian mixture model to cluster the resulting positions. Despite its simplicity in estimation and a reasonable match between the estimated latent positions and clustering labels, some valuable information from the cluster structure which may be needed for the latent position estimations are lost as the estimation of the positions and the cluster model are not done simultaneously. To address this, the second method explores the fully Bayesian estimation of the LPCM using MCMC sampling. This approach estimates both the latent positions and the mixture model parameters simultaneously providing better results but being computationally more expensive.

The latent position samples are assumed to be drawn from a spherical Gaussian mixture as follows

$$X_i \sim \sum_{g=1}^{G} \lambda_g \mathrm{MVN}_d \left(\mu_g, \sigma_g^2 \mathrm{I}\right)$$

where λ_g is the probability that a vertex is in group g, satisfying $\sum_{g=1}^{G} \lambda_g = 1$, and I is the $d \times d$ identity matrix. This structure enables the highly connected vertices to be clustered.

Gorin et al. (2010) extended LPCM to give more flexibility by letting actor covariates contribute to the network organization in various ways, producing an ample family of network models via a mixture of experts modeling schemes. That is, generalized linear model mixtures where the probability of cluster assignment is modeled using a logistic function of covariates. Conversely, Salter-Townshend and Murphy (2013) proposed an alternative inference approach to MCMC sampling of LPCM via variational Bayesian methods to reduce the high computational intensity.

LPCM is attractive in clustering vertices as the underlying latent model automatically gives mixture component means, or means of clustering vertices, as well as the probability of vertex assignments. However, there still exists a major issue of inferring the number of components in the latent mixture distribution. Handcock et al. (2007) suggested using the Bayesian information criterion (BIC) for different potential choices and choosing a number of components with the highest BIC value. However, this approach appears to be computationally expensive to compute the maximum log likelihood required in BIC. Recently, Friel et al. (2013) attempted to tackle this issue by employing conjugate prior distributions which permit almost all latent mixture parameters to be integrated out. This consequently reduces posterior distribution to a more condensed form where the allocation vector of the mixture model still remains, resulting in a fixed dimensional parameter space for trans-model inference. Following the work of Handcock et al. (2007), this model was also constructed using a logistic regression model where the probability of an edge between a vertex pair depends on the Euclidean distance of their respective latent positions. This approach not only escapes from estimating several model parameters as required in Handcock et al. (2007) but also allows for a faster computation time in a larger network relative to the standard techniques available for LPCM.

2.3.5.1 Dot Product Graph Models

This thesis focuses heavily on the random dot product graph model (RDPG) introduced by Nickel (2006) and Young and Scheinerman (2007) which is a type of a latent position random graph model. Similar to LSM, the motivation behind this model is based on the notion that people with common interests are likely to form relationships. Each vertex is associated with a latent vector and the presence or absence of all edges are independent, given the latent vectors in the graph. The probability of an edge between two vertices, conditioned on their latent vectors, is determined by a link function (Section 2.3.5); in this case the dot product of the corresponding latent vectors.

As an example of application, in a social network where vertices denote people and edges denote their friendships, the relative interest of people in different topics may be captured by the components of the vectors whilst the magnitude of the vector captures the talkativeness of the people. Thus, the more talkative the people, the more likely they are to form relationships.

For a distribution, F, on a set, $\mathcal{X} \subset \mathbb{R}^d$, satisfying $\langle x, y \rangle \in [0,1]$ for all $x, y \in \mathcal{X}$. A random graph, G, with associated adjacency matrix, A, for $X = [X_1, \ldots, X_n]^\top \in \mathcal{X}_d$ where

$$\mathcal{X}_d = \{Z \in \mathbb{R}^{n \times d} : ZZ^\top \in [0,1]^{n \times n}, \operatorname{rank}(Z) = d\},$$

is an RDPG distributed according to F, $(X, A) \sim \text{RDPG}(F)$, if

$$\mathbb{P}[A \mid X] = \prod_{i<j} \langle X_i, X_j \rangle^{A_{ij}} (1 - \langle X_i, X_j \rangle)^{1-A_{ij}}.$$

Hence, for the RDPG model, each vertex i has an associated latent position, X_i, and given the latent positions, X_i and X_j, the edges, $A_{ij}|X_i, X_j \overset{ind}{\sim} \text{Bern}(\langle X_i, X_j \rangle)$.

It imperative to note that non-indentifiability is an intrinsic property of RDPGs, akin to many LSMs. In particular, for any orthogonal matrix, $W \in \mathbb{R}^{d\times d}$, and any matrix, X, the dot product between any rows, i and j, of X is indistinguishable from the dot product between the rows, i and j, of XW. Thus, for any distribution, F, on $\mathcal{X}$ and unitary operator U, the distributions $RDPG(F)$ and $RDPG(F \circ U)$ are identical.

The $n \times n$ edge probability matrix is defined by the outer product, $P = XX^\top$, where P is symmetric, positive semidefinite and has a rank of at most d (i.e. $\text{rank}(P) = \text{rank}(X)$). By representing P this way, the spectral properties of P can be easily understood. Specifically, an element of the equivalence class of X is recoverable by performing the spectral decomposition of P, to be discussed in Section 2.4.

Since the birth of the RDPG, innumerable papers have been published that analyze this model in many directions. These include Sussman et al. (2012a) who proposed a spectral embedding procedure (Section 2.4) motivated by the RDPG to assign vertices into blocks in an SBM random graph. They also established a new way to define SBM as an RDPG representation which was later employed in a myriad of papers, for instance Lyzinski et al. (2013), Athreya et al. (2015), and Fishkind et al. (2013a) as well as in this thesis. Shortly after, Sussman et al. (2012b) showed consistency of results in the latent position estimates under the RDPG, while Athreya et al. (2015) explored the aforementioned embedding technique within the RDPG realm, and subsequently Tang et al. (2014) performed a nonparametric hypothesis testing on a pair of RDPGs.

2.4 Adjacency Spectral Graph Embedding

When analyzing latent position random graph models the first step is often to estimate the latent positions, and the estimated latent positions can then be used to perform subsequent analysis. Obtaining accurate estimates of the latent positions will consequently give rise to accurate inference as the latent vectors determine the distribution of the random graph (Sussman et al., 2012a).

For an RDPG defined in Section 2.3.5.1, we have the $n \times n$ edge probability matrix, $P = XX^\top$, that is symmetric, positive semidefinite and has a rank of at most d. Thus, P has a spectral decomposition given by

$$P = [U_P|\widetilde{U}_P][S_P \oplus \widetilde{S}_P][U_P|\widetilde{U}_P]^\top,$$

where $[U_P|\widetilde{U}_P] \in \mathbb{R}^{n \times n}$, $U_P \in \mathbb{R}^{n \times d}$ has orthonormal columns, and $S_P \in \mathbb{R}^{d \times d}$ is a diagonal matrix with non-negative, non-increasing entries along the diagonal. It follows that there exists an orthonormal basis, $W_n \in \mathbb{R}^{d \times d}$, such that $U_P S_P^{1/2} = XW_n$. This introduces obvious non-identifiability as mentioned in Section 2.3.5.1, since XW_n generates the same distribution over adjacency matrices (i.e. $(XW_n) * (XW_n)^\top = XX^\top$). As such, without loss of generality, we consider *uncentered* principal components (UPCA) of X, $\widetilde{X}$, such that $\widetilde{X} = U_P S_P^{1/2}$.

Based on the observation that an adjacency matrix, A, is in essence a noisy form of P (Athreya et al., 2015), the estimate of the UPCA of X is thus the adjacency spectral graph embedding (ASGE) of A (or simply the embedding of A) to dimension d defined below.

Let $A = U_A S_A U_A^\top$ be the (full) spectral decomposition of A. Then the estimate of $\widetilde{X}$ is

$$\widehat{X} = \widehat{V}\widehat{S}^{1/2}, \tag{2.2}$$

where $\widehat{S} \in \mathbb{R}^{d \times d}$ is the diagonal sub-matrix of S_A containing the d largest eigenvalues (in magnitude) of A, and $\widehat{V} \in \mathbb{R}^{n \times d}$ is the sub-matrix of U_A whose orthonormal columns are the respective eigenvectors.

In this thesis, the term embedding is used to highlight the fact that this approach gives a representation of each vertex as a vector in finite dimensional Euclidean space instead of an alternative use of "embedding" in graph theory literature.

The eigen-decomposition of an adjacency matrix provides a means of embedding a graph as points in finite dimensional Euclidean space that can capture the underlying graph structure. This technique is analogous to principle components analysis (PCA) where the variation of the data is captured by its low-dimensional representation. This embedding permits a myriad of statistical and machine learning methodolodies for multivariate Euclidean data to be utilized for graph inference (Sussman, 2014). There have been many recent attempts to apply spectral decomposition techniques on an adjacency matrix of random graphs following the work of Sussman et al. (2012a). For instance, Tang et al. (2013b) showed that under the latent position graph model, and for sufficiently large n, the mapping of the out-of-sample vertices is close to its true latent position via the approach of ASGE, while Lyzinski et al. (2013) proved that the ASGE can be used to obtain perfect clustering for the SBM. Most importantly, Athreya et al. (2015) extended the analysis in Sussman et al. (2014) to show a distributional convergence of the residuals between the estimated and true latent positions. They further proved that for an RDPG the latent positions estimated using ASGE should converge in distribution to a multivariate normal mixture.

The results of Athreya et al. (2015) motivate us to demonstrate the utility of an estimate of the multivariate Gaussian mixture as an empirical prior distribution for estimating block membership of vertices in an SBM graph, details of which will be discussed in Chapter 3.

2.5 Gaussian Mixture Model

In general, given the data, $Y = (y_1, \ldots, y_n)$, where y_i is an independent multivariate observation, the likelihood function of a mixture model with K components can be expressed as

$$f_{mix}(\theta_1, \ldots, \theta_K; \rho_1, \ldots, \rho_K \mid Y) = \prod_{i=1}^{n} \sum_{k=1}^{K} \rho_k f_k \left(y_i \mid \theta_k\right), \tag{2.3}$$

where f_k is the density of the kth component in the mixture model with the corresponding parameters, θ_k, and ρ_k is the mixing proportion that an observation belongs to the kth component, and has to satisfy $\rho_k \in [0, 1]$ and $\sum_{k=1}^{K} \rho_k = 1$.

Here, we consider f_k to be the multivariate normal (Gaussian) density, $\mathcal{N}_k$, with its mean, μ_k, and covariance matrix, Σ_k.

The density of the multivariate normal mixture for y_i is

$$\mathcal{N}_k(Y \mid \mu_k, \Sigma_k) = (2\pi)^{-\frac{n}{2}} \left|\Sigma_k\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(y_i - \mu_k\right)^{\top} \Sigma_k^{-1} \left(y_i - \mu_k\right)\right\} \tag{2.4}$$

In this thesis we employ the model-based clustering of Fraley and Raftery (1999) to find a maximum likelihood estimator of the mixture model assuming K is known, given the latent positions. This procedure is required in the initial stage of our proposed models to derive the empirical Bayes prior for the latent positions in Chapters 3, 4 and 5. This can simply be achieved using the available `R` package `MCLUST` and can be accessed from `http://www.stat.washington.edu/mclust`.

One of the available methods in MCLUST for estimating maximum likelihood clustering with parametrized Gaussian mixture models is the iterative Expectation-Maximization (EM) algorithm, the approach preferred in this study.

Data drawn from multivariate normal mixtures are distinguished by clusters or groups centered at the mean, μ_k. Constant density will give rise to the ellipsoidal surface. The geometric features of the clusters, namely shape, volume, and orientation, are controlled by the covariances, Σ_k. This can be parametrized in order to enforce cross-cluster constraints. For instance, if $\Sigma_k = \lambda I$, then all groups' features will be spherical with equal size, whereas $\Sigma_k = \Sigma$ will give the same geometric features but not necessary spherical, or allowing Σ_k to be unconstrained.

In multivariate Gaussian mixtures, Banfield and Raftery (1993) laid the general groundwork for geometric cross-cluster constraints through parametrization of covariance matrices via eigen-decomposition.

Let A_k be a diagonal matrix whose entries are the eigenvalues, D_k be the orthogonal matrix of the corresponding eigenvectors, and λ_k be a constant of proportionality. Covariance matrix, Σ_k, for the kth component can be eigen-decomposed as

$$\Sigma_k = \lambda_k D_k A_k D_k^T, \tag{2.5}$$

This enables us to treat A_k, λ_k, and D_k independently. Thus, we can either permit these sets of parameters to vary between clusters or fix them to be the same for all clusters. Clusters will have certain common geometric characteristics depending on which parameters are being fixed. More explicitly, A_k controls the shape of the kth component, while λ_k determines its volume of the ellipsoid, and D_k controls its orientation. Fraley and Raftery (2002) stated the three most common models–(1) set $\Sigma_k = \lambda I$, same volume and spherical variance, (2) $\Sigma_k = \lambda_k I$, which clusters will have different volumes but spherical, and (3) $\Sigma_k = \lambda_k A_k$, where the clusters' geometric features are allowed to vary with diagonal covariances.

When clustering latent vectors with more than one dimension, MCLUST provides abbreviations for various geometric features of the model. For instance, EVI signifies a model where the volumes of all clusters are equal (E), allowing the shapes to vary (V), and the orientation is the identity (I). As for this thesis, we mainly use the VVV model which allows all the geometric characteristics to be unconstrained.

2.6 Bayesian Inference

The Bayesian approach begins as early as a traditional frequentist analysis. Given a vector of unknown parameters, θ, and the observed data, $Y = (y_1, \ldots, y_n)$, a sampling model of Y is often given in the form of a probability distribution, $f(Y|\theta)$.

When considered as a function of θ instead of Y, this distribution is commonly referred to as the likelihood, and is sometimes written as $\mathcal{L}(\theta; Y)$ to emphasize the role reversal of θ and Y. Drawing inference about θ usually involves finding the maximum likelihood estimate (MLE) for θ in the frequentist statistical analysis methods. However, in the Bayesian perspective, θ is treated as a random quantity rather than a fixed (unknown) parameter. This can be implemented by adopting a probability distribution for θ which captures any prior information, called the *prior* distribution, $\pi(\theta)$. Inference about θ is then based on its *posterior* distribution which is obtained using Bayes theorem, giving

$$\begin{aligned} p(\theta \mid Y) = \frac{p(Y,\theta)}{p(Y)} &= \frac{p(Y,\theta)}{\int p(Y,\theta) d\theta} \\ &= \frac{f(Y \mid \theta)\pi(\theta)}{\int f(Y \mid \theta)\pi(\theta) d\theta}. \end{aligned} \tag{2.6}$$

The primary issue in implementing the Bayesian framework is the estimation of the posterior distribution which often involves computing the normalizing integral (the denominator in Eqn (2.6)). Further evaluation of integrations may also be required for additional summarization to obtain, for instance, marginal densities, or marginal moments, etc. Thus, it is unavoidable to evaluate the required integrals (Bernardo and Smith, 2009). For simple non-hierarchical Bayesian models the calculation of the integrals can easily be avoided with the appropriate choice of conjugate prior distributions. Nonetheless, with an increase in the dimensionality of the parameter, θ, efficient numerical integration techniques are commonly used.

One of the crucial techniques that facilitates the development of the Bayesian method is Markov chain Monte Carlo (MCMC) for posterior sampling. A method effectively drawing a sequence of values of θ from a proposal distribution and subsequently adjusting these samples for a better estimation of the target posterior distribution, $p(\theta \mid y)$. For any t, the distribution of θ^t conditional on all previous draws, $\theta^1, \theta^2, \ldots, \theta^{t-1}$, solely depends on the most recent value drawn, θ^{t-1}. Thus, the samples, θ's, form a Markov chain. Although the key success to this method is not the Markov property itself but rather that the approximate distribution becomes better as the simulation progresses or converges to a unique stationary distribution where the Markov property plays a role in proving this convergence (Gelman et al., 2014). The acceptance/rejection criteria for each draw is implemented by comparing consecutive states over the target distribution to ensure that the stationary distribution is indeed the posterior distribution.

An appealing feature of MCMC is that the target distribution only needs to be proportional to the posterior distribution; thus avoiding the need to evaluate the integral on the denominator in Eqn (2.6) or a normalizing constant and enabling the exploration of previously computationally impossible applications. Moreover, most applied Bayesians have preferred MCMC methods, as not only do these methods give a fuller complete information of the joint parameter uncertainty from the entire joint posterior distribution, but also they are comparatively easy to implement, despite the very high-dimensional of the models (Carlin and Louis, 2011).

These MCMC techniques are composed of traditional non-iterative methods such as importance sampling (Geweke, 1989), simple rejection sampling, and weighted bootstrap (Smith and Gelfand, 1992). The interested reader is referred to Carlin and Louis (2011) and papers cited therein on these methods. More powerful techniques are the iterative Monte Carlo methods including the *Metropolis–Hastings* (M-H) algorithm (Metropolis et al., 1953; Hastings, 1970), and the *Gibbs sampler* (Geman and Geman, 1984; Gelfand and Smith, 1990). As these algorithms will be executed in this thesis for sampling from the posterior we will discuss them in more detail in Sections 2.6.2 and 2.6.3, respectively.

2.6.1 Empirical Bayes

As mentioned in Section 2.6, Bayesian analysis not only comprises of the likelihood but also a prior distribution for the model parameters. This prior can either be parametric or nonparametric depending on unknown parameters that can subsequently be distributed according to second–stage prior distributions. This ordering of parameters and priors establishes a hierarchical model that must break at a certain point, with all remaining prior parameters assumed known. Rather than imposing this assumption, the empirical Bayes (EB) approach uses the observed data to estimate these final stage prior parameters and proceeds as if previously known.

Consider the two-stage hierarchical model where $f(Y \mid \theta)$ is a likelihood for the observed data Y given a vector of unknown parameters, θ. A prior for θ with density or mass function is given by $\pi(\theta \mid \eta)$, where η is a vector of hyperparameters. For a known η, the posterior distribution according to the Bayes' theorem in Eqn (2.6) can be expressed as follows:

$$p(\theta \mid Y, \eta) = \frac{f(Y \mid \theta)\pi(\theta \mid \eta)}{m(Y \mid \eta)}, \tag{2.7}$$

where $m(Y \mid \eta)$ is the marginal distribution of Y,

$$m(Y \mid \eta) = \int f(Y \mid \theta)\pi(\theta \mid \eta)d\theta. \tag{2.8}$$

In the case where η is unknown, the fully Bayesian approach would place a hyperprior distribution, $h(\eta)$, and compute the posterior distribution as

$$p(\theta \mid Y) = \frac{\int f(Y \mid \theta)\pi(\theta \mid \eta)h(\eta)d\eta}{\int\int f(Y \mid \mathrm{u})\pi(\mathrm{u} \mid \eta)h(\eta)d\mathrm{u}d\eta} = \int p(\theta \mid Y, \eta)h(\eta \mid Y)d\eta. \tag{2.9}$$

In EB analysis, the marginal distribution as expressed in Eqn (2.8) is instead employed to estimate η by $\widehat{\eta} \equiv \widehat{\eta}(Y)$, for instance, the marginal maximum likelihood estimator. Inference is then dependent on the estimated posterior distribution, $p(\theta \mid y, \widehat{\eta})$. The integration in the rightmost part of Eqn (2.9) is thus substituted by a maximization in the EB approach which highly reduces a computational complexity. The term "empirical Bayes" derives from the fact that the data is being used to estimate the hyperparameter, η.

2.6.2 Metropolis-Hasting Sampler

The Metropolis sampler was initially developed by Metropolis et al. (1953) and later generalized by Hastings (1970) in which simulations following his method are known as the Metropolis-Hastings sampler. In Bayesian inference, it is an MCMC method that samples a sequence of random variables from a probability distribution (i.e. the posterior) that is relatively difficult to sample from, particularly when the posterior is complex. In addition, the Metropolis sampler has another interesting property that adds to its appeal, this being that the target distribution, $p(\theta \mid y)$, only needs to be known up to the proportionality constant. Thus the calculation of the normalizing integral is not required.

Suppose we wish to draw the multivariate θ vector from a joint posterior distribution i.e. $p(\theta \mid Y) \propto g(\theta) \equiv f(Y \mid \theta)\pi(\theta)$, where g represents the unnormalized posterior. The sampler begins with an initial point, $\theta^{(0)}$, and then, for each $h \in \{1, 2, 3, \dots, \}$, generating a candidate point, θ^*, from a proposal distribution, $q(\cdot \mid \theta^{(h-1)})$, which depends only on the previous point, $\theta^{(h-1)}$.

The metropolis algorithm imposes the symmetry condition on q (i.e. $q(x \mid y) = q(y \mid x)$). Subsequently, θ^h is then set to either θ^* or $\theta^{(h-1)}$ depending on the acceptance ratio, r. That is,

$$r = \frac{g(\theta^*)}{g(\theta^{h-1})}. \tag{2.10}$$

Hastings (1970) later generalized the algorithm revoking the symmetry requirement on q, resulting in the ratio, r, in Eqn (2.10) to be replaced by

$$r = \frac{g(\theta^*)q(\theta^{(h-1)} \mid \theta^*)}{g(\theta^{(h-1)})q(\theta^* \mid \theta^{(h-1)})}, \tag{2.11}$$

which is now commonly known as the Metropolis-Hastings (M-H) sampler.

To summarize, the M-H sampler associated with target distribution, g, and proposal distribution, q, is given in Algorithm 1.

Algorithm 1 The Metropolis-Hasting Algorithm

```
Initialization: Choose an arbitrary starting value $\theta^{(0)}$
At iteration:    $h$ $(h \geqslant 1)$;

Given: $\theta^{(h-1)}$,
Generate: $\theta^* \sim q(\theta \mid \theta^{(h-1)})$

Compute the acceptance probability :
    $pi(\theta^{(h-1)}, \theta^*) = \min \left\{ 1, \frac{g(\theta^* \mid y)q(\theta^{(h-1)} \mid \theta^*)}{g(\theta^{(h-1)} \mid y)q(\theta^* \mid \theta^{(h-1)})} \right\}.$

Set:
    $\theta^{(h)} = \begin{cases} \theta^* & \text{with probability } \pi(\theta^{(h-1)}, \theta^*), \\ \theta^{(h-1)} & \text{with probability } 1 - \pi(\theta^{(h-1)}, \theta^*). \end{cases}$
```

Note that in Section 3.2.1 a special case of Algorithm 1 is employed where the proposal distribution, q, is independent of the current state of the chain; this is known as an independent M-H algorithm.

2.6.3 Gibbs Sampler

The Gibbs sampler, a special case of the M-H algorithm (Section 2.6.2), was proposed by Geman and Geman (1984) without prior knowledge of previously existing work. The paper by Geman and Geman (1984) focused more on optimization for finding the posterior mode rather than simulation. As a result, it took a considerable amount of time for the Bayesian community to understand that the posterior distribution can be simulated by using the Gibbs sampler scheme and that most Bayesian inference can be accomplished by MCMC. This was not obvious until the publication by Gelfand and Smith (1990). In a Gibbs update, the proposal is from a conditional distribution of the target distribution where random samples generated are always accepted. The sampler uses the conditional distribution of one component of the state vector given the rest of the components which are known as "full conditionals".

Consider the parameter vector, Θ, has been partitioned into d sub-vectors, $\Theta = (\theta_1, \ldots, \theta_d)$. At iteration t, each $\theta_j^{(t)}$ is sampled from the conditional distribution given all the other components of Θ sequentially:

$$p(\theta_j |\, \theta_{-j}^{(t-1)}, y),$$

where $\theta_{-j}^{(t-1)}$ denotes all the components of Θ, excluding θ_j, at their current values:

$$\theta_{-j}^{(t-1)} = \left(\theta_1^{(t)}, \ldots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, \ldots, \theta_d^{(t-1)}\right).$$

The Gibbs sampler scheme is summarized in Algorithm 2 below.

Algorithm 2 Gibbs Update

```
Initialization: Choose an arbitrary starting value $\Theta^{(0)} = \left(\theta_1^{(0)}, \ldots, \theta_d^{(0)}\right)$
At iteration:    $h$ $(h \geqslant 1)$;

1. Generate: $\theta_1^{(h)} \sim p(\theta_1^{(h)} |\, \theta_2^{(h-1)}, \ldots, \theta_d^{(h-1)})$

2. Generate: $\theta_2^{(h)} \sim p(\theta_2^{(h)} |\, \theta_2^{(h)}, \theta_3^{(h-1)}, \ldots, \theta_d^{(h-1)})$
   ⋮

j. Generate: $\theta_j^{(h)} \sim p(\theta_j^{(h)} |\, \theta_1^{(h)}, \ldots, \theta_{j-1}^{(h)}, \theta_{j+1}^{(h-1)}, \ldots, \theta_d^{(h-1)})$
   ⋮

d. Generate: $\theta_d^{(h)} \sim p(\theta_d^{(h)} |\, \theta_1^{(h)}, \ldots, \theta_{d-1}^{(h)})$
```

As such, each sub-vector, θ_j, is updated conditionally on the latest values of the other components of θ, namely the iteration t values of the previously updated components and the iteration $t-1$ values for the others.

The Gibbs sampler will be revisited in Sections 3.2.1, 4.3.2, and 5.3.1 for sampling from a posterior where conditional distributions are obtainable.

In this chapter notations and terminologies of network/graphs were introduced. This was followed by a sizable body of relevant literature reviews on statistical random graph models, the adjacency spectral graph embedding method and its recent theoretical advances, as well as Bayesian inference. These subjects will lay the groundwork for the methodologies proposed in Chapters 3, 4, and 5.

3 | EMPIRICAL BAYES ESTIMATION FOR THE STOCHASTIC BLOCKMODEL

Statistical inference on graphs is a rapidly growing field in many areas of science, engineering, and business. As discussed in Section 1.2, community detection is an important task in network analysis, where vertices who are partitioned into the same group exhibit a more similar connectivity pattern amongst themselves than with other vertices outside the group. Discovering groups of vertices that exhibit similar attributes and/or structural roles within the graph not only allows us to identify groups of interest for a deeper understanding of a network, but also to predict future or unknown connectivity patterns by exploring a condensed form of the network's overall structure (Aicher et al., 2014). This chapter presents a new approach, motivated by recent theoretical advances on ASGE (defined in Section 2.4), that provides an empirical Bayes methodology for community detection.

A popular probabilistic graph model that finds inherent community structures is the stochastic blockmodel (SBM). A common approach for clustering the vertices in a graph under the SBM is to estimate the block memberships of all vertices. Although the task of estimating the block memberships has generated an interesting body of research in various contexts over the years, this remains challenging. As mentioned previously in Section 2.3.4, many techniques have been proposed to accomplish this task, namely likelihood maximization (Bickel and Chen, 2009; Choi et al., 2012; Celisse et al., 2012; Bickel et al., 2013), maximization of modularity (Newman, 2006), spectral techniques (Rohe et al., 2011; Sussman et al., 2012a; Fishkind et al., 2013b) and Bayesian methods (Snijders and Nowicki, 1997; Nowicki and Snijders, 2001; Handcock et al., 2007; Airoldi et al., 2008).

In parallel to the SBM, the notion of representing a network by associating a latent vector/-position in a $d-$dimensional Euclidean space to each vertex is introduced by Hoff et al. (2002) (see Section 2.3.5). In this model, vertices that are in close proximity are more inclined to link than vertices that are distant in the latent space. A special case of the latent position model which we will focus on in this thesis is the RDPG (Section 2.3.5.1). The RDPG model assumes that, conditioned on the latent vectors, the presence of an edge is an independent Bernoulli trial whose success probability is given by the dot product of the two adjoining latent vectors. As an example, in the context of social networking with vertices denoting members in the network, the latent vectors and the dot product semantically capture the perception of differing levels of "interests" and "talkativeness" (Young and Scheinerman, 2007).

Recently, Athreya et al. (2015) proved that for an RDPG, the latent positions estimated using ASGE converge in distribution to a multivariate Gaussian mixture. This result motivates the treatment of the estimated latent positions (embeddings) of a K-block SBM graph as (approximately) an independent and identically distributed sample from a mixture of K multivariate Gaussians. In this chapter, we demonstrate the utility of an estimate of this multivariate Gaussian mixture as an empirical prior distribution for estimating block memberships in an SBM graph, as this empirical prior quantifies residual uncertainty in the proposed model parameters after ASGE. To do so, we will represent an SBM graph as an RDPG graph. While the SBM depends on an inherently non-geometric construction, as in each block is assigned to a categorical label in which it governs the adjacency probability, the RDPG model depends on a geometric construction where each block is assigned to a point in Euclidean space (i.e. a latent vector). The dot product of these latent vectors then governs the adjacency probability in the graph. Thus, by parametrizing the standard SBM as the RDPG model whose spectral properties of the adjacency matrix are well-understood, has been proven beneficial in the analysis and clustering of the adjacency spectral graph embedding (Sussman et al., 2012a).

Section 3.1 presents the SBM as an RDPG. In Section 3.2 we present the proposed empirical Bayes model, introduce alternative models to compare with the proposed model, and give details of the Metropolis-within-Gibbs algorithm for posterior sampling from the model. Section 3.3 briefly discusses a model non-identifiability issue and how we deal with it. We then carry out simulations to assess the performance of the algorithm and describe the results obtained in Section 3.4. Finally, Section 3.5 discusses further extensions to the model, provides a summary and concludes the chapter.

3.1 Setting and Main Theorem

Following the notation outlined in Section 2.2, we consider simple undirected graphs so that the adjacency matrix, A, is symmetric, hollow (no self-loops implies $A_{ii} = 0$ for all i), and binary (no multi-edges or weights implies $A_{ij} \in \{0, 1\}$ for all i, j). For our random graphs, the vertex set is fixed; it is the edge set that is random.

Recall from Section 2.3.5.1 that a random graph, G, with adjacency matrix, A, is said to be a random dot product graph (RDPG) if

$$f(A \mid X) = \prod_{i<j} \langle X_i, X_j \rangle^{A_{ij}} (1 - \langle X_i, X_j \rangle)^{1-A_{ij}}. \tag{3.1}$$

Thus, in the RDPG model each vertex, $i \in V$, is associated with a latent vector, $X_i \in \mathbb{R}^d$, and the probability of an edge existing between two vertices, i and j, is given by

$$\mathbb{P}\left(i \sim j\right) = \langle X_i, X_j \rangle.$$

Conditional on the latent vectors, presence/absence of edges are independent Bernoulli random variables, $A_{ij}|X_i, X_j \overset{ind}{\sim} \text{Bern}(\langle X_i, X_j \rangle)$. Clearly, the latent vectors must satisfy the probability constraints, i.e. $0 \leq \langle X_i, X_j \rangle \leq 1$.

The SBM can be formally defined as an RDPG for which all vertices that belong to the same block share a common latent vector according to the following definition.

Definition 3.1 ((Positive Semidefinite) Stochastic Blockmodel)**.** An RDPG can be parametrized as an SBM with K blocks if the number of distinct rows in X is K. That is, let the probability mass function, f, associated with the distribution, F, of the latent positions, X_i, be given by the mixture of point masses

$$f(X_i \mid \rho, \nu) = \sum_{k=1}^{K} \rho_k \delta_{\nu_k}(X_i), \tag{3.2}$$

where the block membership probability vector, $\rho \in (0,1)^K$, satisfies $\sum_{k=1}^{K} \rho_k = 1$ and the distinct latent positions are represented by $\nu = [\nu_1 | \cdots | \nu_K]^\top \in \mathbb{R}^{K \times d}$. Thus, the standard definition of the SBM defined in Section 2.3.4, that is SBM(K, ρ, B) with parameters, K, ρ, and $B = \nu\nu^\top$, is seen to be an RDPG with $X_i \mid \rho, \nu \overset{iid}{\sim} \sum_k \rho_k \delta_{\nu_k}$.

Consequently, we have

$$\mathbb{P}\left(i \sim j\right) = B_{\tau_i, \tau_j} = \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle = \langle X_i, X_j \rangle, \tag{3.3}$$

where τ is the block membership vector and B is the block probability matrix.

In this setting, the block memberships, $\tau_1, \ldots, \tau_n \mid K, \rho \overset{iid}{\sim} \text{Discrete}([K], \rho)$, such that $\tau_i = \tau_j$ if and only if $X_i = X_j$. Let N_k be the number of vertices such that $\tau_i = k$; we will condition on

$N_k = n_k$ throughout.

In what follows, we assume that the block probability matrix, B, has distinct rows; that is, $B_{k,\cdot} \neq B_{k',\cdot}$ for all $k \neq k'$, to circumvent the non–identifiability issues or enforce the affinity assumption for the SBM (see Section 3.3 below).

Remark 3.2. The above definition can be conceived as a special case of a classic SBM where it is often formalized by the block probability matrix, B, together with the block proportions, ρ, as previously stated in Section 2.3.4. However, the two definitions coincide on condition that B is positive semidefinite.

To date, Bayesian approaches for estimating the block memberships in the SBM have typically involved a specification of the prior on the matrix, $B = \nu\nu^\top$; the beta distribution (which includes the uniform distribution as a special case) is often chosen as the prior (Snijders and Nowicki, 1997; Nowicki and Snijders, 2001). Facilitated by our re-casting of the SBM as an RDPG and recent theoretical advances described in Section 3.2.1 below, we will instead derive an empirical prior for the latent positions, ν, themselves.

3.2 Models

One of the many benefits of the Bayesian inference method is that prior knowledge can be integrated, thus permitting a fuller account of the uncertainties associated with the parameters. This section presents the theoretical aspects and algorithms of the proposed model which will be employed to investigate the utility of our empirical Bayes model for estimating block memberships in an SBM graph, dubbed *ASGE* and presented in Section 3.2.1.

For comparison purposes, in Sections 3.2.2 and 3.2.3 we construct an alternative *Flat* and two benchmark models, namely *Exact* and *Gold*, as outlined below. The names of these models are named after their respective prior distributions used for the latent positions, ν.

- **Flat** – an alternative to the proposed empirical Bayes prior distribution for ν. Since in the absence of the ASGE theory a natural choice for the prior on ν is the uniform distribution.
- **Exact** – a primary benchmark model where all model parameters, except the block membership vector, τ, is assumed known.
- **Gold** – a secondary benchmark model where ν and τ are the unknown parameters; the gold standard mixture of Gaussians prior distribution for ν takes its hyperparameters to be the

true latent positions and theoretical limiting covariances motivated by the distributional results from Athreya et al. (2015) presented below.

3.2.1 The Empirical Bayes with ASGE Prior (“**ASGE**”)

This section details the proposed empirical Bayes estimation model for estimating block memberships of vertices in an SBM graph. Within this section, the likelihood, prior and posterior structures for the proposed model are discussed, and the posterior sampling algorithm for this model is also suggested. This is followed by a graphical representation to further illustrate the posterior distribution in Section 3.2.1.1.

Recall that the mechanism used in this thesis that provides a way to embed a graph as points in finite dimension Euclidean space is the eigen-decomposition of an adjacency matrix (i.e. the ASGE technique). More specifically, as defined in Eqn (2.2), we estimate the latent positions, X, by $\widehat{X} = \widehat{S}^{1/2}\widehat{V}$, where $\widehat{S} \in \mathbb{R}^{d\times d}$ is the diagonal matrix and consists of d largest eigenvalues of A in descending order along the diagonal and $\widehat{V} \in \mathbb{R}^{n\times d}$ is their corresponding eigenvectors.

Recently, Athreya et al. (2015) proved that for an RDPG, the latent positions estimated using ASGE converge in distribution to a multivariate Gaussian mixture. We can express this more formally in a central limit theorem (CLT) for the scaled differences between the estimated and true latent positions of the RDPG graph, as well as a corollary to motivate our empirical Bayes prior (henceforth denoted $ASGE$).

Theorem 3.3 (Athreya et al. (2015))**.** *Let G be an RDPG with d-dimensional latent positions, $X_1, \dots, X_n \overset{iid}{\sim} F$, and assume distinct eigenvalues for the second moment matrix of F. Let $\widetilde{X} \in \mathbb{R}^{n\times d}$ be the (uncentered) principal components version of X so that the columns of $\widetilde{X}$ are orthogonal. Let $\mathcal{N}(0, \Sigma)$ represent the cumulative distribution function for the multivariate normal, with mean, 0, and covariance matrix, Σ. Then for each row $\widetilde{X}_i$ of $\widetilde{X}$ and $\widehat{X}_i$ of $\widehat{X}$,*

$$\sqrt{n}(\widetilde{X}_i - \widehat{X}_i) \xrightarrow{\mathcal{L}} \int \mathcal{N}(0, \Sigma(x))dF(x),$$

where the integral denotes a mixture of the covariance matrices and, with $\Delta = \mathbb{E}[X_1 X_1^\top]$,

$$\Sigma(x) = \Delta^{-1}\mathbb{E}[X_j X_j^\top(x^\top X_j - (x^\top X_j)^2)]\Delta^{-1}. \tag{3.4}$$

The special case of the SBM gives rise to the following corollary.

Corollary 3.4. *In the setting of Theorem 3.3, suppose G is an SBM with K blocks. Then, if we condition on $X_i = \nu_k$, we obtain*

$$P\left(\sqrt{n}\left(\widehat{X}_i - \nu_k\right) \leq z \middle| X_i = \nu_k\right) \to \Phi(z, \Sigma_k), \tag{3.5}$$

where $\Sigma_k = \Sigma(\nu_k)$ with $\Sigma(\cdot)$ defined as in Theorem 3.3.

Note that the distribution, F, of the latent positions, X, remains unchanged, as $n \to \infty$.

These results motivate the mixture of normals approximation, $\widehat{X}_1, \cdots, \widehat{X}_n \overset{iid}{\sim} \sum_k \rho_k \varphi_k$, for the estimated latent positions obtained from the ASGE where φ_k is multivariate normal density as previously expressed in Eqn (2.4). More precisely, these embeddings can be considered as (approximately) an independent and identically distributed sample from a mixture of multivariate Gaussians.

A similar Bayesian method for latent position clustering of network data is proposed in Handcock et al. (2007). As previously mentioned in Section 2.3.5, their LPCM is an extension of Hoff et al. (2002), wherein all the key features of network data are incorporated simultaneously – namely clustering, transitivity, and homophily on attributes. Under their model, latent vector is assumed to follow a finite Gaussian mixture distribution where a cluster of vertices is represented by each component. The LPCM is similar to our model, but they use the logistic function instead of the dot product as their link function.

Theorem 3.3 gives rise to a method for obtaining an empirical prior for ν using the ASGE. Given the estimated latent positions, $\widehat{X}_1, \ldots, \widehat{X}_n$, obtained via the spectral embedding of the adjacency matrix, A, (see Section 2.4), the next step is to cluster these $\widehat{X}_i$ using Gaussian mixture models (GMM) as described in Section 2.5. There are a wealth of methods available for this task; as for this chapter we employ the model-based clustering of Fraley and Raftery (2002). This mixture estimate, in the context of Corollary 3.4, quantifies our uncertainty about the latent position vector, ν, suggesting its role as an empirical Bayes prior distribution. Our empirical Bayes prior distribution for ν can be expressed as

$$f(\nu \mid \widehat{\mu}, \widehat{\Sigma}) \propto \mathbb{I}_{\mathcal{S}}(\nu) \prod_{k=1}^{K} \mathcal{N}_d(\nu_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k), \tag{3.6}$$

where $\mathcal{N}_d(\nu_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k)$ is the density function of a multivariate normal distribution with mean, $\widehat{\mu}_k$, and covariance matrix, $\widehat{\Sigma}_k$, denoting standard maximum likelihood estimates (via Expectation-Maximization algorithm) based on the embeddings, $\widehat{X}_i$, and has the density with dimension, d, as

$$\mathcal{N}_d(\nu_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k) = (2\pi)^{-\frac{n}{2}} \left|\widehat{\Sigma}_k\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\nu_k - \widehat{\mu}_k\right)^T \widehat{\Sigma}_k^{-1} \left(\nu_k - \widehat{\mu}_k\right)\right\}.$$

The indicator, $\mathbb{I}_{\mathcal{S}}(\nu)$, in Eqn (3.6) enforces homophily and block identifiability constraints for the SBM via

$$\mathcal{S} = \{\nu \in \mathbb{R}^{K\times d} : 0 \leq \langle \nu_k, \nu_{k'} \rangle \leq \langle \nu_k, \nu_k \rangle \leq 1 \ \forall k, k' \in [K] \ \text{ and } \ \langle \nu_k, \nu_k \rangle > \langle \nu_{k'}, \nu_{k'} \rangle \ \forall k > k'\}. \tag{3.7}$$

An overview of the procedure for obtaining empirical Bayes prior using the ASGE and GMM is amalgamated in Algorithm 3 below.

Algorithm 3 Empirical Bayes estimation using the ASGE empirical prior

```
Given graph $G$
Obtain adjacency spectral embedding $\widehat{X}$
Obtain empirical prior via GMM of $\widehat{X}$
Sample from the posterior via Metropolis–within–Gibbs (see Algorithm 4 below)
```

In the setting of Corollary 3.4, for an adjacency matrix, A, the likelihood for the block membership vector, $\tau \in [K]^n$, and the latent positions, $\nu \in \mathbb{R}^{K\times d}$, is given by

$$f(A \mid \tau, \nu) = \prod_{i<j} \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle)^{1-A_{ij}}. \tag{3.8}$$

This is the case where the block memberships, τ, the latent positions, ν, are assumed unknown. We assume, for the prior distribution, that the latent positions, ν, is independent of τ. We choose conditionally independent multinomial distributions for the components of τ with hyperparameter, ρ, where it is chosen to follow a Dirichlet distribution with parameters $\theta_k = 1$ for all k in the unit simplex, Δ_K. The prior distribution for the latent vectors, ν, is appointed to be a truncated multivariate normal distribution with mean, $\widehat{\mu}_k$, and a covariance matrix, $\widehat{\Sigma}_k$. These are estimated from fitting a K-component GMM using `MCLUST` on the embeddings, $\widehat{X}_i$, motivated by Athreya et al. (2015)'s theorem (Theorem 3.3). Rather than specifying hyperprior distributions for both μ_k and Σ_k, we instead employ $\widehat{\mu}_k$ and $\widehat{\Sigma}_k$ given by the GMM fit, $\widehat{X}_i$; hence the empirical prior. It is clear that the name of the proposed model, *ASGE*, is coined from the procedure used to estimate the hyperparameters (i.e. μ_k and Σ_k) from the ASGE technique.

To sum up, the prior distributions on the model parameters, τ, ν, and ρ, are

$$\begin{aligned}\tau \mid \rho &\sim \text{Multinomial}(\rho),\\ \rho &\sim \text{Dirichlet}(\theta),\\ \nu \mid \widehat{\mu}, \widehat{\Sigma} &\sim \mathbb{I}_{\mathcal{S}}(\nu) \prod_{k=1}^{K} \mathcal{N}_d(\nu_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k).\end{aligned}$$

Our empirical posterior for the *ASGE* model is then relatively straightforward and is given by

$$f(\tau, \nu, \rho \mid A) \propto f(A \mid \tau, \nu) \cdot f(\tau \mid \rho) \cdot f(\rho \mid \theta) \cdot f(\nu \mid \widehat{\mu}, \widehat{\Sigma}),$$

By choosing a conjugate Dirichlet prior for ρ, $f(\rho \mid \theta) = \text{Dirichlet}(\theta)$ for $\theta \in \Delta_K$, we can marginalize the posterior distribution over ρ as follows:

$$\begin{aligned}f(\tau, \nu \mid A) &= \int_{\Delta_K} f(\tau, \nu, \rho \mid A)\, d\rho\\ &\propto f(A \mid \tau, \nu) f(\nu \mid \widehat{\mu}, \widehat{\Sigma}) \int_{\Delta_K} f(\tau \mid \rho) f(\rho \mid \theta)\, d\rho.\end{aligned}$$

Let $T = (T_1, \ldots, T_K)$ denote the block assignment counts, where $T_k = \sum_{i=1}^{n} \mathbb{I}_{\{k\}}(\widehat{\tau}_i)$. Then the resulting prior distribution is given by

$$\begin{aligned}f(\tau \mid \theta) = \int_{\Delta_K} f(\tau \mid \rho) f(\rho \mid \theta) d\rho &= \frac{\Gamma(\sum_{k=1}^{K} \theta_k)}{\prod_{k=1}^{K} \Gamma(\theta_k)} \int_{\Delta_K} \left(\prod_{i=1}^{n} \rho_{\tau_i}\right) \left(\prod_{k=1}^{K} \rho_k^{\theta_k - 1}\right) d\rho\\ &= \frac{\Gamma(\sum_{k=1}^{K} \theta_k)}{\prod_{k=1}^{K} \Gamma(\theta_k)} \int_{\Delta_K} \underbrace{\prod_{k=1}^{K} \rho_k^{\theta_k + T_k - 1}}_{\propto \text{Dirichlet}(\theta + T)} d\rho\\ &= \frac{\Gamma(\sum_{k=1}^{K} \theta_k)}{\prod_{k=1}^{K} \Gamma(\theta_k)} \frac{\prod_{k=1}^{K} \Gamma(\theta_k + T_k)}{\Gamma(n + \sum_{k=1}^{K} \theta_k)},\end{aligned}$$

which follows a Multinomial-Dirichlet distribution with parameters θ and n. Therefore, the marginal posterior distribution can be expressed as

$$\begin{aligned}f(\tau, \nu \mid A) &\propto f(A \mid \tau, \nu) \cdot f(\tau \mid \theta) \cdot f(\nu \mid \widehat{\mu}, \widehat{\Sigma})\\ &\propto f(A \mid \tau, \nu) \cdot \left[\prod_{k=1}^{K} \Gamma(\theta_k + T_k)\right] \cdot f(\nu \mid \widehat{\mu}, \widehat{\Sigma}).\end{aligned}$$

We can sample from the marginal posterior distribution for τ and ν via Metropolis–Hasting–within–Gibbs sampling.

Algorithm 4 Metropolis–Hasting–within–Gibbs sampling for the block membership vector, τ, and the latent positions, ν.

```
At iteration $t$;
Gibbs step:
for $i = 1$ to $n$ do
    Compute $\rho_i^*\left(\tau_1^{(t)}, \dots, \tau_{i-1}^{(t)}, \tau_{i+1}^{(t-1)}, \tau_n^{(t-1)}\right)$ as in Eqn (3.10)
    Set $\tau_i^{(t)} = k$ with probability $\rho_{i,k}^*$
end for
M–H step:
for $m = 1$ to 10 do
    Propose $\widetilde{\nu} \sim \mathbb{I}_{\mathcal{S}}(\nu) \prod_{k=1}^{K} \mathcal{N}_d(\widetilde{\nu}_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k)$
    Compute the acceptance probability $\pi(\widetilde{\nu}) = \min\left\{1, \frac{f(A|\tau^{(t)},\widetilde{\nu})}{f(A|\tau^{(t)},\nu^{(t-1)})}\right\}$
    Set
        $$\nu^{(t)} = \begin{cases} \widetilde{\nu} & \text{with probability } \pi(\widetilde{\nu}) \\ \nu^{(t-1)} & \text{with probability } 1 - \pi(\widetilde{\nu}) \end{cases}$$
end for
```

A standard Gibbs sampling update is employed to sample the posterior of τ, which can be updated sequentially. The idea behind this method is to first consider a full conditional posterior distribution of τ. Let $\tau_{-i} = \tau \setminus \tau_i$ denote the block memberships for all but vertex, i. Then, conditioning on τ_{-i}, we have

$$f(\tau_i \mid \tau_{-i}, A, \nu, \theta) \propto \prod_{j \neq i} \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle)^{1-A_{ij}} \cdot \left[\prod_{k=1}^{K} \Gamma(\theta_k + T_k) \right]. \tag{3.9}$$

Hence, the posterior distribution for $\tau_i \mid \tau_{-i}, A, \nu, \theta \sim \text{Multinomial}(\rho_{\mathrm{i}}^*)$ where

$$\rho_{i,k}^* = \frac{\Gamma(\theta_k + T_k) \prod_{j \neq i} \langle \nu_k, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_k, \nu_{\tau_j} \rangle)^{1-A_{ij}}}{\sum_{k'=1}^{K} \Gamma(\theta_{k'} + T_{k'}) \prod_{j \neq i} \langle \nu_{k'}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{k'}, \nu_{\tau_j} \rangle)^{1-A_{ij}}}. \tag{3.10}$$

The procedure consists of visiting each τ_i for $i = 1, \dots, n$ and executing Algorithm 4. Recently, Decelle et al. (2011) suggest a regime where it is statistically possible to optimally infer the exact block memberships of vertices given that the initial values of the model parameters are the correct ones. However, as our purpose is to demonstrate the advantages of our empirical Bayes methodology, it is more appealing to initialize τ with $\tau^{(0)} = \widehat{\tau}$, the block assignment vector obtained from the GMM clustering of the embeddings, $\widehat{X}$.

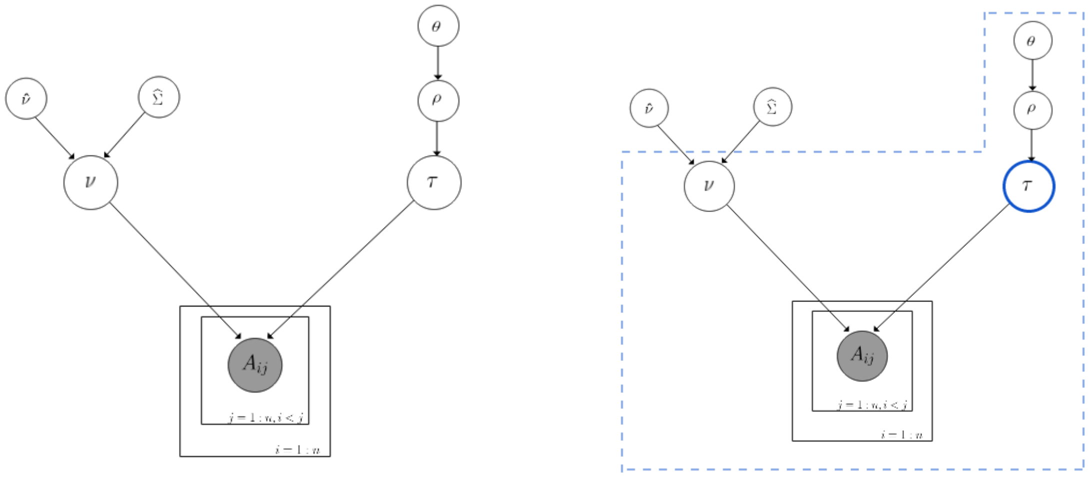
Unfortunately, the full conditional posterior distribution of ν is not of standard form such that we can generate from it directly. As such, an independent M-H algorithm is utilized as addressed in Section 2.6.2 to update ν. For the M-H sampler for ν, the prior distribution, $f(\nu \mid \widehat{\mu}, \widehat{\Sigma})$, as expressed in Eqn (3.6) will be employed as the proposal distribution, q. We generate a proposed state, $\widetilde{\nu} \mid \widehat{\mu}, \widehat{\Sigma} \sim f(\nu \mid \widehat{\mu}, \widehat{\Sigma})$, with the acceptance probability defined as

$$\min \left\{ \frac{f(A \mid \tau, \widetilde{\nu})}{f(A \mid \tau, \nu)}, 1 \right\},$$

where ν in the denominator denotes the current state. The initialization of ν is $\nu^{(0)} \mid \widehat{\mu}, \widehat{\Sigma} \sim f(\nu \mid \widehat{\mu}, \widehat{\Sigma})$. Note that since the proposal distribution for ν is the prior distribution, the M-H acceptance probability is thus defined by the ratio of the likelihood functions (for the proposed and previously accepted values).

3.2.1.1 Directed Acyclic Graph Representation

With interest growing in studying graphical models (Jordan, 2004) within the fields of computer science and statistics, using a directed acyclic graph (DAG) to represent Bayesian hierarchical models is becoming a common practice. DAG is an appealing tool for visualizing conditional independence between random variables or a joint probability distribution, and has been popular for Bayesian inference as early as the 1990s. Figure 3.1 displays the *ASGE* model's hierarchical structure.



(A) DAG of the *ASGE* model.

(B) Markov blanket of τ_i

Figure 3.1: DAGs

In this case, the vertices in the graph signify random variables, the absence of edges signifies conditional independence among the random variables in the model, and the direction of an arrow signifies a *parent–child* relationship. Thus, each vertex is independent from the rest of the graph except for its descendants, conditioning on its parents. When sampling using an MCMC scheme, particularly Gibbs sampling which requires the computation of the full conditionals (Section 2.6.3), the notion of Markov blankets (Pearl, 2014) of graphical models is of advantage in this setting. The Markov blanket of a vertex consists of a set of its parents, co-parents and children which separates it from the rest of the graph. As an example, in the *ASGE* model the implementation of the Gibbs sampling scheme to update each element of block memberships, τ, would require the full conditional density as expressed in Eqn (3.9), the Markov blanket of τ_i can be achieved by

$$\begin{aligned} f\left(\tau_i \mid \text{rest}\right) &= f\left(\tau_i \mid \text{Markov blanket of } \tau_i\right) \\ &= f\left(\tau_i \mid \tau_{-i}, A, \theta, \nu, \rho\right) \\ &\propto f\left(\tau_i \mid \tau_{-i}, A, \theta, \nu\right) \\ &= f\left(A \mid \tau, \nu\right) \cdot f\left(\tau \mid \theta\right) \\ &= \prod_{j \neq i} \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle)^{1-A_{ij}} \left[\prod_{k=1}^{K} \Gamma(\theta_k + T_k) \right], \end{aligned}$$

where ‘rest’ denotes all the vertices in the graph except τ_i. See Figure 3.1b for a graphical representation where the Markov blanket is shown using dashed lines.

3.2.2 The Alternative “**Flat**” Model

In the event that no special prior information is available, a natural choice of prior is the uniform distribution. This results in the formulation of the *Flat* model as an alternative to an empirical Bayes prior distribution for ν on the constraint set, $\mathcal{S}$, as stated in Eqn (3.7), where the marginal posterior distribution for τ and ν is given by

$$\begin{aligned} f(\tau, \nu \mid A) &\propto f(A \mid \tau, \nu) \cdot f(\tau \mid \theta) \cdot f(\nu) \\ &\propto f(A \mid \tau, \nu) \left[\prod_{k=1}^{K} \Gamma(\theta_k + T_k) \right] \mathbb{I}_{\mathcal{S}}(\nu), \end{aligned}$$

where the likelihood, $f(A \mid \tau, \nu)$, is defined in the same manner as the *ASGE* model, disclosed in Eqn (3.8). Again, the Gibbs sampler for τ is identical to the procedure presented in Section 3.2.1. As for the M-H sampler for the latent positions, ν, the flat prior distribution is used as the proposal. However, we initialize ν by generating it from the prior distribution of ν as in the *ASGE* model, i.e. $f(\nu \mid \widehat{\mu}_k, \widehat{\Sigma}_k)$.

3.2.3 Comparison Benchmarks

"Exact"

The *Exact* model is constructed as our primary benchmark where the latent positions, ν, and the block membership probabilities, ρ, are assumed known. Thus, it is only necessary to specify prior distribution for the block membership, τ, and it is assumed to follow a multinomial distribution with the parameter vector, ρ, i.e. $\tau \sim \text{Multinomial}(\rho)$.

Thus, the posterior distribution of the block membership, τ, is given by

$$\begin{aligned} f(\tau \mid A, \nu, \rho) &\propto f(A \mid \tau, \nu) \cdot f(\tau \mid \rho) \\ &= \prod_{i=1}^{n} \rho_{\tau_i} \prod_{i<j} \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle)^{1-A_{ij}}. \end{aligned} \tag{3.11}$$

We can draw inferences about τ based on the posterior, $f(\tau \mid A, \nu, \rho)$, via an *Exact* Gibbs sampler. Then, conditioning on τ_{-i}, we have

$$f(\tau_i \mid \tau_{-i}, A, \nu, \rho) \propto \rho_{\tau_i} \prod_{j \neq i} \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle)^{1-A_{ij}}. \tag{3.12}$$

where the posterior distribution for $\tau_i \mid \tau_{-i}, A, \nu, \rho \sim \text{Multinomial}(\rho_i^*)$ with

$$\rho_{i,k}^* = \frac{\rho_k \prod_{j \neq i} \langle \nu_k, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_k, \nu_{\tau_j} \rangle)^{1-A_{ij}}}{\sum_{k'=1}^{K} \rho_{k'} \prod_{j \neq i} \langle \nu_{k'}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{k'}, \nu_{\tau_j} \rangle)^{1-A_{ij}}}. \tag{3.13}$$

Hence, for the *Exact* Gibbs sampler, once a vertex is selected, the exact calculation of ρ_i^* is available and sampling from the Multinomial(ρ_i^*) is straightforward. Initialization of τ will be $\tau_1^{(0)}, \ldots, \tau_n^{(0)} \mid \rho \overset{iid}{\sim} \text{Multinomial}(\rho)$.

"Gold"

For our secondary benchmark model, we assume that the block membership probability vector, ρ, is known but both ν and τ are unknown. In order to obtain a posterior distribution for τ and ν given the data, A, a prior distribution on the latent position, ν, is required. Here we describe what we call the *Gold* prior distribution.

Let the true value for the latent positions be represented by ν^*. Based on Corollary 3.4, we can suppose that the prior distribution for ν follows a (truncated) multivariate Gaussian centered at

ν^* and with covariance matrix, $\Sigma^* = (1/n)\Sigma$, given by the theoretical limiting distribution for the embedding, $\widehat{X}$, presented in Eqn (3.4). This corresponds to the approximate distribution of $\widehat{X}_i$ if we condition on $\tau_i = k$. Hence, the *Gold* standard prior distribution. To summarize, the prior distributions on the model parameters, τ and ν, are

$$\tau \mid \rho \sim \text{Multinomial}\,(\rho)\,,$$
$$\nu \mid \nu^*, \Sigma^* \sim \mathbb{I}_{\mathcal{S}}(\nu) \prod_{k=1}^{K} \mathcal{N}_d\left(\nu_k \mid \nu_k^*, \Sigma_k^*\right),$$

where the constraints, $\mathcal{S}$, on ν is previously stated in Eqn (3.7).

Inference for τ and ν is based on the posterior distribution, $f(\tau, \nu \mid A, \rho)$, estimated by samples obtained from a Gibbs sampler for τ, and an independent M-H sampler for ν. Thus, the joint posterior distribution for the unknown quantities is given by

$$\begin{aligned} f(\tau, \nu \mid A, \rho) &\propto f(A \mid \tau, \nu) \cdot f(\tau \mid \rho) \cdot f(\nu \mid \nu^*, \Sigma^*) \\ &= \left[\prod_{i=1}^{n} \rho_{\tau_i} \prod_{i<j} \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{A_{ij}} (1 - \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle)^{1-A_{ij}} \right] f(\nu \mid \nu^*, \Sigma^*). \end{aligned} \tag{3.14}$$

The Gibbs sampler for τ in this case is identical to that for the *Exact* model except the initial state, $\tau^{(0)}$, is given by the estimated block assignment vector, $\widehat{\tau}$, obtained from the GMM clustering result of the estimated latent positions, $\widehat{X}$, as described in Section 2.5. Similar to *ASGE* in Section 3.2.1, the prior distribution, $f(\nu \mid \nu^*, \Sigma^*)$, will be employed as the proposal distribution for the M-H sampler for ν.

Table 3.1 provides a summary of our Bayesian prior specification schemes. The ASGE theory suggests that we might expect increasingly better performance from *Flat* to *ASGE* to *Gold* to *Exact*.

Table 3.1: Bayesian Sampling Schemes

Model			**Exact**	**Gold**	**ASGE**	**Flat**
Parameter			a primary benchmark	a secondary benchmark	the proposed model	an alternative to the *ASGE* model
Gibbs	$\boldsymbol{\tau_i}$	Prior	$\tau_i \vert \rho \sim \text{Multinomial}(\rho)$	$\tau_i \vert \rho \sim \text{Multinomial}(\rho)$	$T \vert \theta \sim \text{Multinomial} - \text{Dirichlet}(\theta, n)$	$T \vert \theta \sim \text{Multinomial} - \text{Dirichlet}(\theta, n)$
		Initial Point	$\tau_i \vert \rho \sim \text{Multinomial}(\rho)$	$\widehat{\tau}$	$\widehat{\tau}$	$\widehat{\tau}$
Independent Metropolis Hasting	$\boldsymbol{\nu_k}$	Prior	–	$\nu_k \vert \nu_k^*, \Sigma_k^* \sim \mathcal{N}(\nu_k^*, \Sigma_k^*)$	$\nu_k \vert \widehat{\mu}_k, \widehat{\Sigma}_k \sim \mathcal{N}(\widehat{\mu}_k, \widehat{\Sigma}_k)$	$\nu_k \sim \mathcal{U}(\mathcal{S})$
		Initial point	–	$\nu_k^{(0)} \vert \nu_k^*, \Sigma_k^* \sim \mathcal{N}(\nu_k^*, \Sigma_k^*)$	$\nu_k^{(0)} \vert \widehat{\mu}_k, \widehat{\Sigma}_k \sim \mathcal{N}(\widehat{\mu}_k, \widehat{\Sigma}_k)$	$\nu_k^{(0)} \vert \sim \mathcal{N}(\widehat{\mu}_k, \widehat{\Sigma}_k)$
		Proposal	–	$\widetilde{\nu}_k \vert \nu_k^*, \Sigma_k^* \sim \mathcal{N}(\nu_k^*, \Sigma_k^*)$	$\widetilde{\nu}_k \vert \widehat{\mu}_k, \widehat{\Sigma}_k \sim \mathcal{N}(\widehat{\mu}_k, \widehat{\Sigma}_k)$	$\widetilde{\nu}_k \sim \mathcal{U}(\mathcal{S})$

3.3 Model Identifiability

Bayesian analysis for finite mixture models has been investigated by many researchers since the emergence of the MCMC techniques. Mixture models, as discussed in Section 2.5, provide a flexible way of dealing with heterogeneous data. Particularly, these models give a natural groundwork for statistical modelling where data is presumed to belong to one of the K classes or components and that individual class memberships are unknown. In a Bayesian paradigm, while MCMC provides an easy way to draw inference from the posterior distribution involving mixture models, there are issues arising from the MCMC. One such issue is the non-identifiability of the components, or what is commonly referred to as a "label-switching problem" in the MCMC output.

The posterior distribution will be invariant to permutations of the mixture model's parameters labeling if we place exchangeable priors on them. Thus, for each mixture component we will have identical marginal posterior distributions of the parameters. This implies that throughout MCMC simulation the sampler experiences the switches of the labels as a consequence of the symmetries of the posterior distribution. It is therefore pointless to infer or summarize straight from MCMC output using ergodic averaging over labels.

As label switching of the components is a pre-requisite of MCMC convergence in Bayesian mixture models, this must be resolved first. A number of solutions have surfaced in the literature over the past years to tackle this issue including artificial identifiability constraints (Richardson and Green, 1997), relabelling algorithms (Stephens and Phil, 1997; Stephens, 2000; Celeux, 1998), and the decision theoretic approach by defining label invariant loss functions (Celeux et al., 2000; Hurn et al., 2003). Jasra et al. (2005) provides an excellent overview of the available solutions to the label switching problem.

For our empirical Bayes model, to deal with the aforementioned problem, we place restrictions on the parameters as suggested by Richardson and Green (1997). Thus, block identifiability constraints, Eqn (3.7), are defined as a condition on the parameter space. Since this identifiability constraint does not originate from any belief about the model, but instead is an inferential convenience, it is thus 'artificial'. This method works for us because the likelihood is invariant to the labeling of the parameters so the symmetry can be broken in the prior (and thus posterior), and thus solve the label switching problem.

3.4 Simulation Study

We illustrate the performance of the *ASGE* model via various Monte Carlo simulation experiments and one real data experiment. Specifically, Section 3.4.1 illustrates a two-block SBM case, while a generalization of this two-block SBM to a more general RDPG is discussed in Section 3.4.2. Moreover, Section 3.4.3 depicts a three-block SBM case, and lastly we consider a three-class Wikipedia graph example in Section 3.4.4. We demonstrate the utility of the *ASGE* model for estimating vertex block assignments via comparison to its benchmark models and competing methods including *Exact*, *Gold*, *Flat*, and GMM.

To aid the understanding of the implementation of the *ASGE* model, a two-block SBM case in Section 3.4.1 is further broken down into three subsections. First, Section 3.4.1.1 provides a detailed description of MCMC implementation, followed by a discussion on convergence assessment in Section 3.4.1.2, and lastly Section 3.4.1.3 shows the performance of the proposed model by comparing it to other alternative methods.

3.4.1 A Simulation Example with $K = 2$

Consider the SBM parameterized by

$$B = \begin{pmatrix} 0.42 & 0.42 \\ 0.42 & 0.5 \end{pmatrix} \qquad \text{and} \qquad \rho = (0.6, 0.4). \tag{3.15}$$

The block membership probability vector, ρ, indicates that each vertex will be in block 1 with probability $\rho_1 = 0.6$ and in block 2 with probability $\rho_2 = 0.4$. Edge probabilities are determined by the entries of B, independent and a function of only the vertex block memberships. This model can be parameterized as an RDPG in $\mathbb{R}^2$ where the probability mass function, f, of the latent positions as defined in Theorem 3.3 is a mixture of point masses,

$$f(X_i \mid \nu, \rho) = \sum_{k=1}^{K} \rho_k \delta_{\nu_k}(X_i),$$

positioned at $\nu_1 \approx (0.5489, 0.3446)$ with prior probability 0.6 and $\nu_2 \approx (0.3984, 0.5842)$ with prior probability 0.4.

3.4.1.1 MCMC Implementation

Inference for our models are relatively straightforward, particularly MCMC implementation for the *ASGE* model can be accomplished by following the steps in Algorithm 4 as discussed in Section 3.2.1.

For each $n \in \{100, 250, 500, 750, 1000\}$, we generate 500 random graphs according to the SBM with parameters as provided in Eqn (3.15). For each graph, G, the spectral decomposition of the corresponding adjacency matrix, A, as outlined in Section 2.4, provides the estimated latent positions, $\widehat{X}$. Subsequently, GMM is used to cluster the (embedded) vertices via `MCLUST` in `R` package using `VVV` model, as described in Section 2.5. The results of which are then employed in the *ASGE* model, where the estimated block memberships, $\widehat{\tau}$, are set to be the initial values in the Gibbs step for updating τ, the mixture component means, $\widehat{\mu}_k$, and covariance matrices, $\widehat{\Sigma}_k$, determine our empirical Bayes *ASGE* prior for the latent positions, ν. To avoid the model selection challenges $d = 2$ and $K = 2$ are assumed known in this experiment.

For the performance analysis, the block membership vector, τ, and the latent positions, ν, are the quantities of interest while the block proportion vector, ρ, may be regarded as a nuisance parameter. Thus, for the *ASGE* and *Flat* model, a conjugate Dirichlet prior is placed on ρ which allows us to integrate it out. This yields a marginal posterior distribution of τ and ν which we can then generate posterior samples of τ and ν from. This procedure is carried out for a large number of iterations for two parallel Markov chains until convergence. The posterior inference for τ is based on iterations after convergence (see below). That is, the last 500 iterations of each of two parallel chains, making a total of 1000 iterations. Performance is then evaluated by calculating the vertex block assignment error. By following the same recipe we provided above, this procedure is repeated 500 times to obtain estimates of the error rates.

3.4.1.2 MCMC Convergence Monitoring

With the implementation of various MCMC techniques, adopting Bayesian methods which often have issues of analytic intractability and computation in-feasibility has now become possible. MCMC computations are amazingly easy and flexible to implement, however there are few issues which cannot be neglected. These include, for instance, how long the Markov chains need to run to achieve convergence, and obtaining enough simulations for a more precise inference (Gelman and Shirley, 2011). As any inferences from MCMC output are mainly dependent on the assumption that the simulated Markov chain has converged or reached its stationary distribution, assessing convergence of MCMC sampling procedures is one of the most crucial tasks (Brooks and Roberts, 1998). Many diagnostic tests have been formulated in the literature for monitoring convergence of MCMC chains. A review of such approaches is compiled in Brooks and Roberts (1998); Cowles and Carlin (1996). The convergence diagnostic developed by Gelman and Rubin (1992) and further addressed by Gelman (1996) has been implemented in this thesis.

In general, given the observations simulated thus far, the Gelman–Rubin method comprises of analyzing multiple independent chains to form a distributional estimate of the variance of random variable of interest (Brooks and Roberts, 1998). It gives an insight of how close a chain is to a steady state and how much improvement of the estimate is expected with more simulations (Brooks and Roberts, 1998).

The approach relies on a simple analysis of variance of scalar quantities of interest, for instance, θ. Suppose the variance of θ is known and labelled as σ^2 under the target distribution, f. If σ^2 could be estimated by $\widetilde{V}$ from the output of the sample, the proportion, $\widetilde{V}/\sigma^2$, gives an estimate of what rate of the total amount of information about θ has been retrieved from the simulations employed to compute $\widetilde{V}$.

Given c independent parallel chains of each scalar summary, θ, each with different initial points that are over-dispersed with respect to the target distribution, f, the estimator, $\widetilde{V}$, is computed by forming a weighted average of the between chain variance means, B/n, and within chain variance, W, for each θ as follows

$$\frac{B}{n} = \frac{1}{c-1}\sum_{i=1}^{c}(\overline{\theta}_i - \overline{\theta})^2,$$

where $\overline{\theta}_i = \frac{1}{n}\sum_{j=n+1}^{2n}\theta_{ij}$ is the j^{th} observation of θ from i^{th} chain, and $\overline{\theta} = \frac{1}{c}\sum_{i=1}^{c}\overline{\theta}_i$ is the average scalar summary across all simulated chains. The mean of variance within the c chains, W is given by

$$W = \frac{1}{c}\sum_{i=1}^{c}\left(\frac{1}{n-1}\sum_{j=n+1}^{2n}(\theta_{ij} - \overline{\theta}_i)^2\right).$$

Then, the estimator, $\widetilde{V}$, is defined as

$$\widetilde{V} = \frac{n-1}{n}W + \left(1 + \frac{1}{c}\right)\frac{B}{n}.$$

The comparison of between and within–chain variances is expressed as the ratio,

$$R = \frac{\widetilde{V}}{\sigma^2}.$$

However, the denominator of R is not known and can be (under)estimated from the data by W in order to have an overestimate R. Thus, the Gelman–Rubin method monitors

$$\sqrt{\widehat{R}} = \sqrt{\frac{\widetilde{V}}{W}},$$

which is called the *potential scale reduction factor* (PSRF). If the PSRF approaches one, this essentially indicates that each of the c chains of n iterations has converged to the target distribution.

For our models, the scalar quantity of interest, θ, defined above is the percentage of misassigned vertices per iteration which is used to compute PSRF to assess convergence of the chains. It is common to choose c to be two or more parallel chains with over-dispersed initial values to ensure that the sample space of the parameters are fully explored as well as to reduce the possibility of the sampling being trapped in a certain area of the sample space (Gelman and Rubin, 1992; Gelman, 1996).

In this research, we set $c = 2$ chains and follow the steps in Algorithm 5 to assess the convergence of MCMC. That is, for two parallel chains we initially generate 1000 iterations from the posterior densities of the block membership, τ, and the latent positions, ν, with independent starting points as previously discussed in Section 3.4.1.1, then dispose the first half and compute the probabilities of misassigned vertices for each iteration, $\widehat{\theta}$. Subsequently, we simulate 1000 bootstrap samples of $\widehat{\theta}$ and calculate Gelman–Rubin statistics (PSRF) for each bootstrap sample. The last step is to assess the convergence of the chains by checking whether 1.1 (a recommended cut-off by Gelman and Rubin (1992)) lies within the 95% bootstrap confidence interval of PSRF, estimated by the bias-corrected and accelerated (BCa) bootstrap of Efron (1987). If it meets this criterion we then combine the last 500 draws of both chains for posterior inference, and we may assume that the convergence has occurred. Otherwise, we continue drawing another 500 iterations and repeat the steps until there is no evidence of non-convergence.

As an illustration of the convergence of MCMC using the Gelman–Rubin statistic, Figure 3.2 gives the $\widehat{R}$ for the parameter chains, θ, of the last 1000 iterations used for posterior inference for the *ASGE* model. This is from one of the graph realizations for various choices of number of

Algorithm 5 MCMC convergence assessment via Gelman–Rubin approach

```
1:  for each of $c = 2$ parallel chains do
2:      Simulate $t = 1000$ iterations from posterior densities of parameters of interest
3:      Discard the first 500 draws
4:      Compute scalar summary statistic $\widehat{\theta} = (\widehat{\theta}, \dots, \widehat{\theta}_t)$
5:      Generate $N = 1000$ bootstrap resamples of $\widehat{\theta}$ and calculate $\widehat{R}_1, \dots, \widehat{R}_N$
6:      if 1.1 lies within the 95% bootstrap confidence interval of $\widehat{R}$ then
7:          Combine the remaining total draws from both chains for inference
8:          return
9:      else
10:         Simulate another 500 iterations, combine with the previous 500 draws and repeat 3-5
            until converged
11:     end if
12: end for
```

vertices, n. It is evident from Figure 3.2 that all the PSRF values obtained are very close to 1 and below the 1.1 threshold which shows signs of convergence. Figure 3.3 shows the number of iteration runs before convergence for the *ASGE* model. It is apparent from the histograms that the larger the graph the higher the number of iterations needed before convergence. Nevertheless, the number of iterations before convergence in the *ASGE* model is still half of the *Flat* model (up to 50,000), where the results are not presented here for brevity. Thus, our *ASGE* model is computationally more efficient than the *Flat* model.

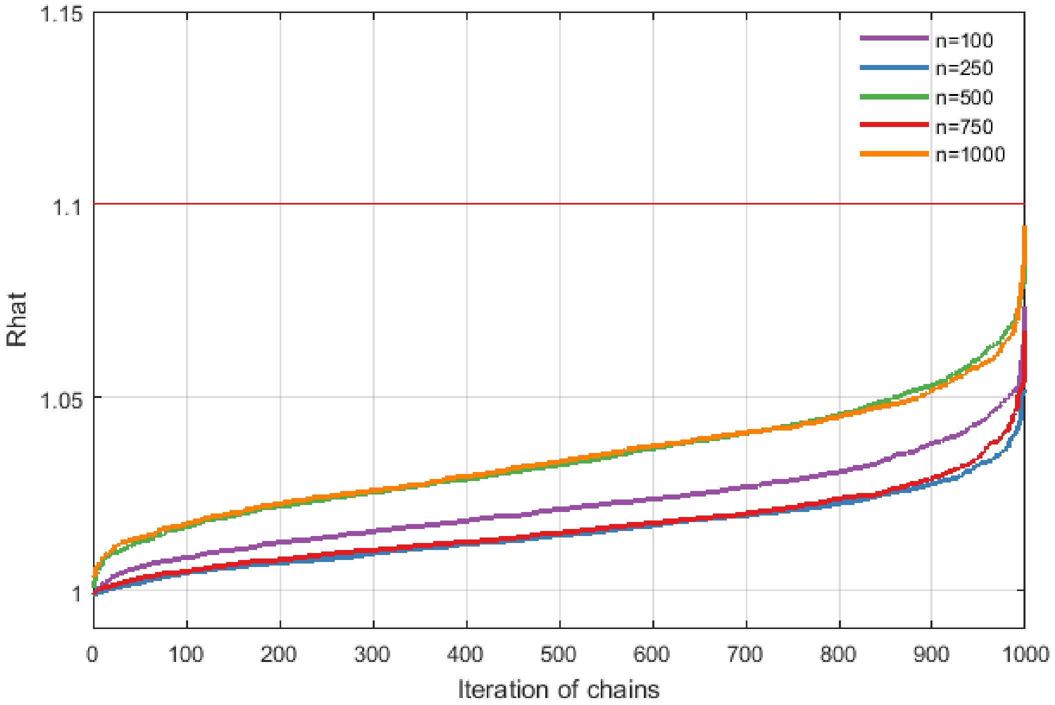
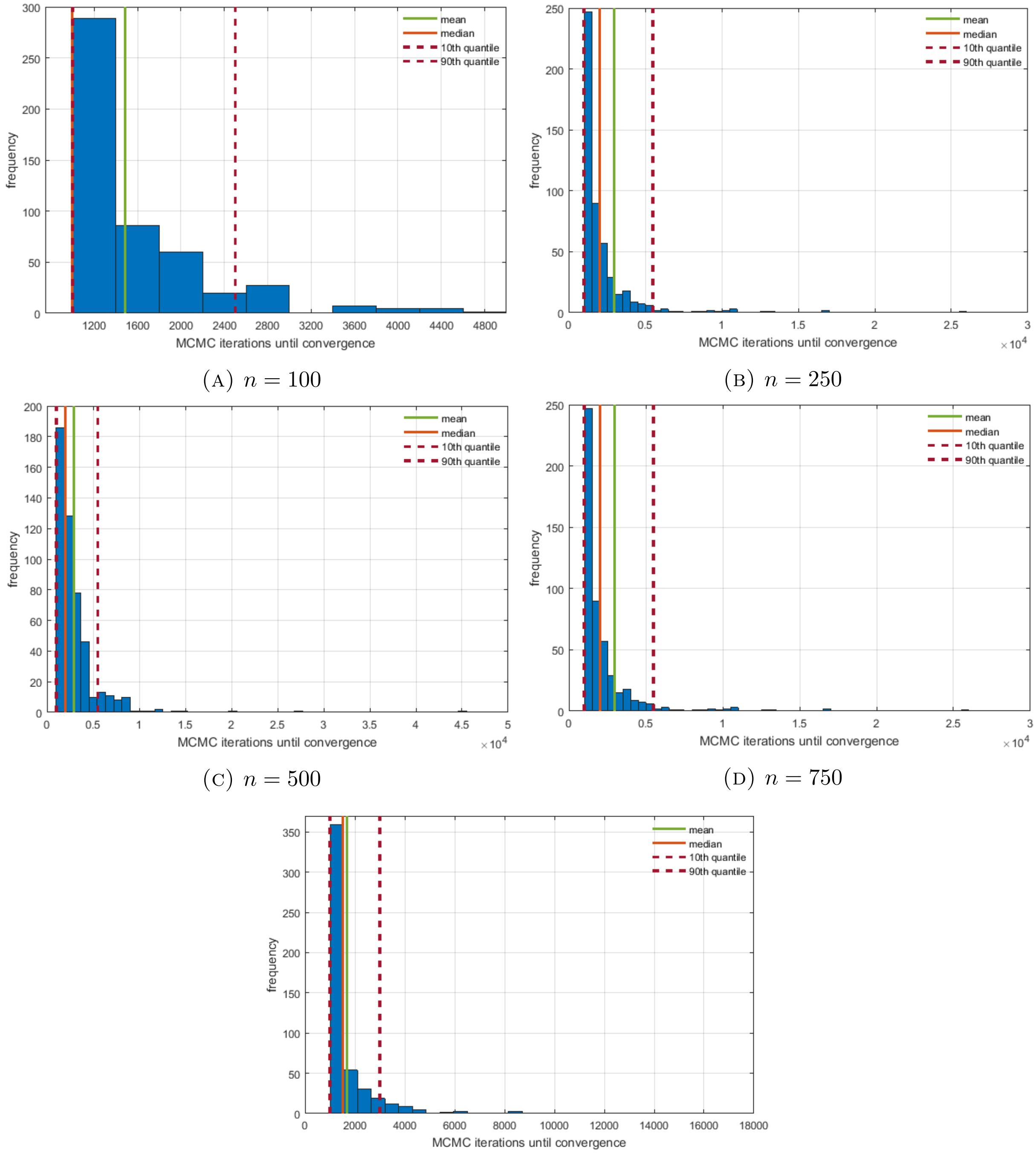


FIGURE 3.2: $\widehat{R}$ over the simulation length for the Gelman–Rubin method for convergence of Markov chains for $n = \{100, 250, 500, 750, 1000\}$ of the *ASGE* model. The red horizontal line indicates the 1.1 threshold for convergence.



(A) $n = 100$

(B) $n = 250$

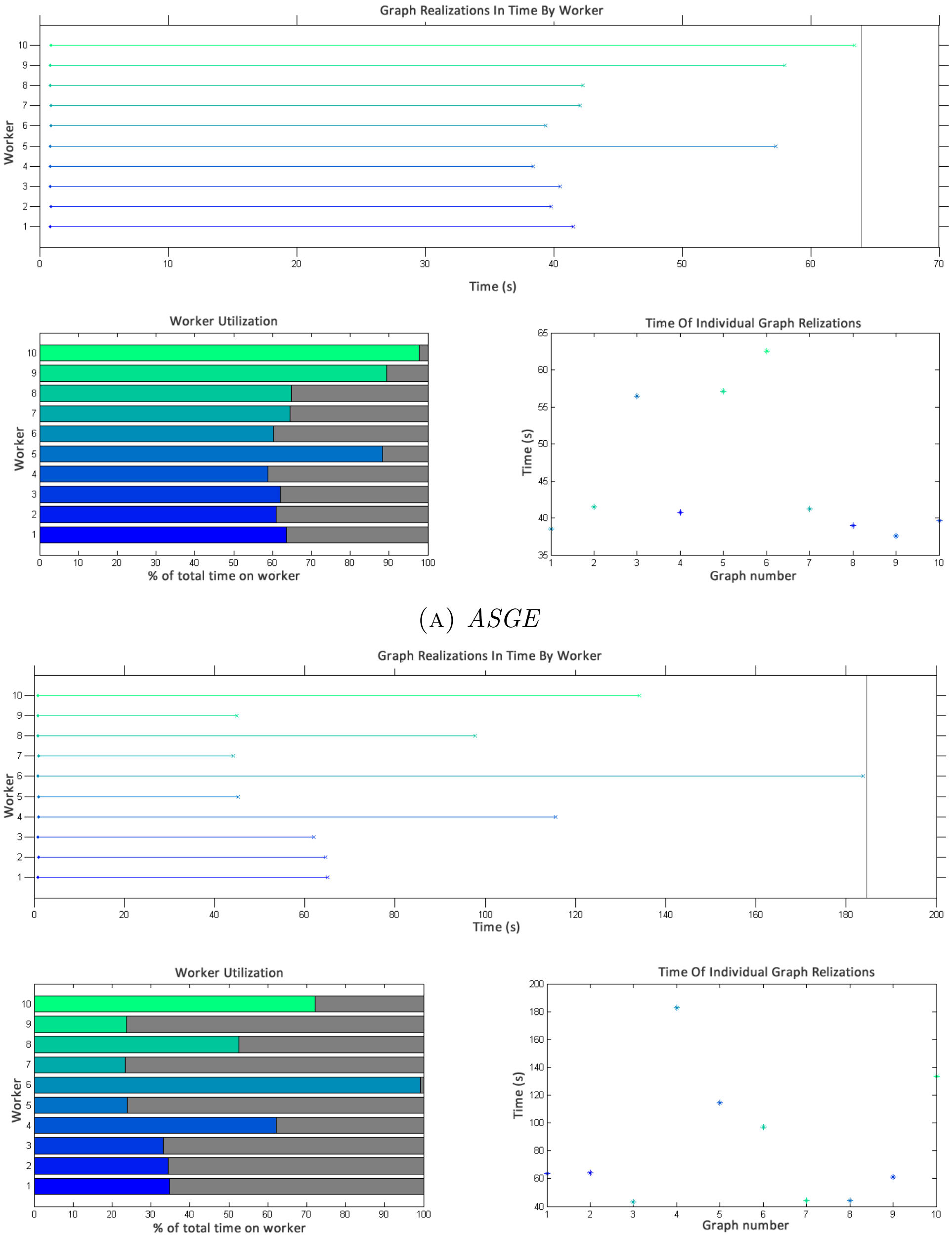
(C) $n = 500$

(D) $n = 750$

(E) $n = 1000$

FIGURE 3.3: Histograms for the total number of MCMC iterations until convergence of the empirical Bayes *ASGE* model for the two-block SBM considered in Section 3.4.1 with a different choice of n.

To affirm this claim, Figure 3.4 displays the runtime in seconds for the simulations of the *ASGE* and *Flat* model on $n = 100$ vertices using the Matlab parallel computing environment (MATLAB, 2015). For instance, based on the 10 graph realizations demonstrated in Figure 3.4, the *ASGE* model on average takes 45.42 seconds whereas it takes 84.84 seconds for the *Flat* model.

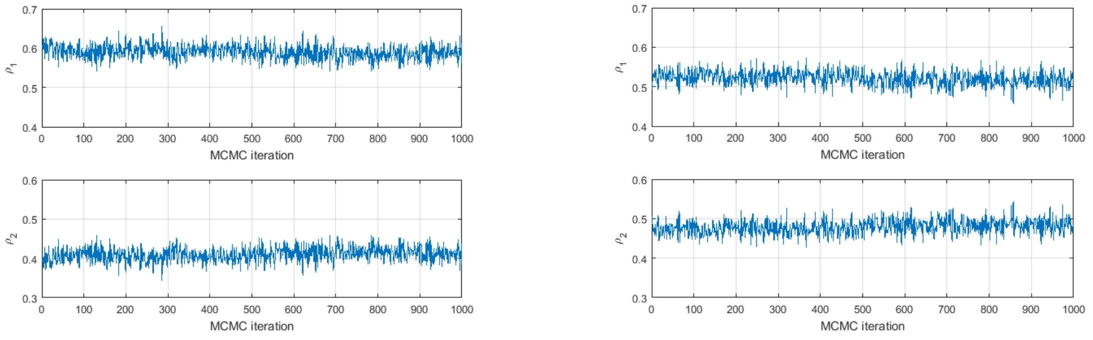


(A) *ASGE*

(B) *Flat*

FIGURE 3.4: Runtime in seconds of the simulations of the *ASGE* and *Flat* model on $n = 100$ for 10 graph realizations using the Matlab parallel toolbox.

Although standard graphical summaries such as trace plots from individual posterior samples do not usually give enough information to be reliable tools for assessing convergence, there is no harm in seeing whether the plots agree with our convergence diagnostic method. Thus, taken together, these outcomes provide a reasonable picture to monitor the convergence of MCMC. Figure 3.5 display trace plots for the *ASGE* and *Flat* model. It is apparent that the MCMC chains move monotonically, thus no evidence of non-convergence was found.

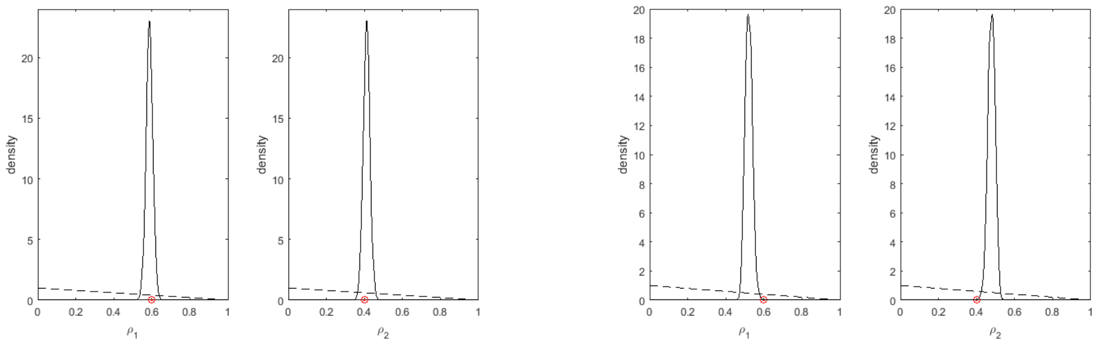


(A) *ASGE*

(B) *Flat*

FIGURE 3.5: Trace plots of block proportions, ρ, for the empirical Bayes *ASGE* and *Flat* models for $n = 1000$.

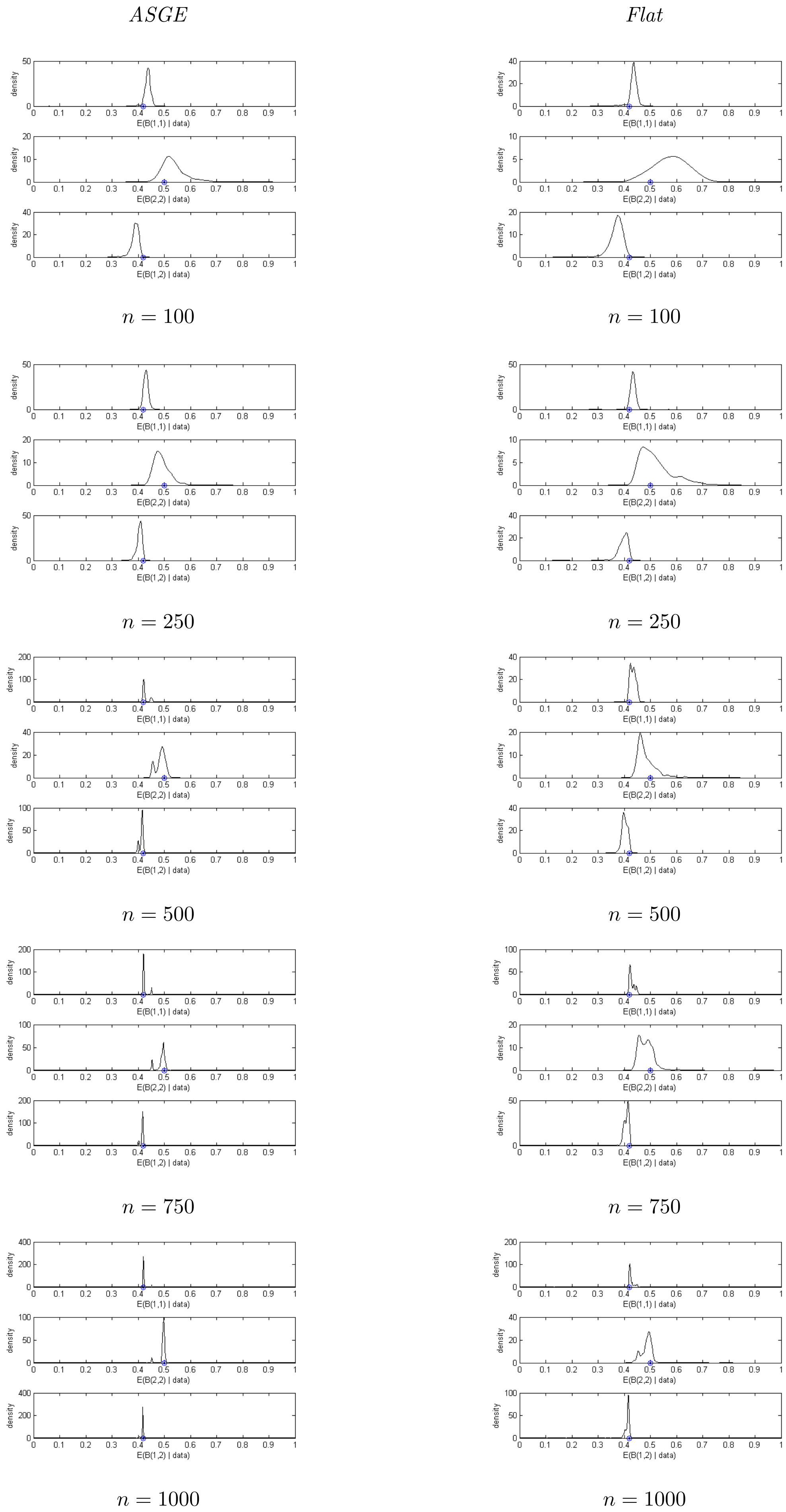
Moreover, inference about the nuisance parameter, ρ, is not required here, although it is interesting to observe their prior and posterior distributions. Figure 3.6 shows the marginal prior and posterior densities using kernel density estimates constructed from the last 1000 MCMC iterations. Here, we adopt a Gaussian kernel density estimator with diffusion–based bandwidth selection from Botev et al. (2010, Algorithm 1). Note that these figures are from a single graph realization of $n = 1000$ for both *ASGE* and *Flat* models. It becomes apparent that the true parameter values which are indicated by red points on the horizontal axis, are centered around the posterior densities for the *ASGE* model, but less so for the *Flat* model. A similar remark can be made from the visual inspection of kernel densities fitted to posterior means for the dot product of the latent vectors associated to each block (i.e. the estimates of matrix B in Eqn (3.15)) illustrated in Figure 3.7.



(A) *ASGE*

(B) *Flat*

FIGURE 3.6: Marginal prior densities (dashed curves) and posterior densities (solid curves) for the block proportion vector ρ of one of the graph realizations for $n = 1000$. Red points on the horizontal axis indicate the true parameter values.



ASGE *Flat*

$n = 100$ $n = 100$

$n = 250$ $n = 250$

$n = 500$ $n = 500$

$n = 750$ $n = 750$

$n = 1000$ $n = 1000$

FIGURE 3.7: Kernel densities fitted to posterior means for the edge probability matrix, B, obtained from 500 graphs for $n = \{100, 250, 500, 750, 1000\}$. Blue points on the horizontal axis indicate the true parameter values. *Left*: the *ASGE* model. *Right*: the *Flat* model.

3.4.1.3 Performance Comparisons

In this section, we evaluate the block membership estimation performance of the *ASGE* model on the two-block SBM synthetic graphs discussed in Section 3.4.1. This is achieved by obtaining the estimates of error rates, $\widehat{\epsilon}$. The procedure begins with the calculation of the vertex block assignment error for each vertex, $i \in V$, across $M = 1000$ iterations by

$$v(i) = 1 - \frac{\sum_{j=1}^{M} \mathbb{I}_{\{\tau_i^*\}} \widehat{\tau}_i^{(j)}}{M}, \tag{3.16}$$

where τ_i^* is the true block label for vertex, i, and $\widehat{\tau}_i^{(\cdot)}$ denotes the posterior block assignment samples for vertex, i. Subsequently, we take the average of the obtained vertex block assignment errors over n vertices from Eqn (3.16) as

$$\bar{\epsilon} = \frac{\sum_{i=1}^{n} v(i)}{n}, \tag{3.17}$$

then the estimate of the error rate for a random experiment over $G \in \{1, \ldots, 500\}$ graph realizations is

$$\widehat{\epsilon} = \mathbb{E}\left[\bar{\epsilon}\right]. \tag{3.18}$$

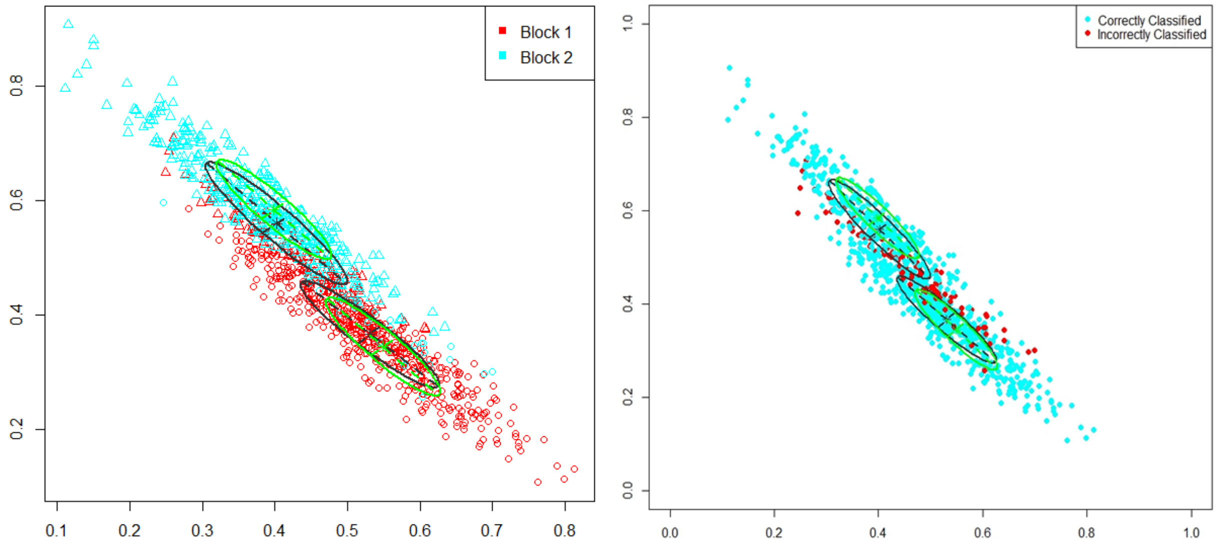


FIGURE 3.8: Scatter plot of the embeddings, $\widehat{X}_i$, for one Monte Carlo replicate with $n = 1000$ for the two-block SBM considered in Section 3.4.1. In the left panel, the colors denote the true block memberships for the corresponding vertices in the SBM, while the symbols denote the cluster memberships given by the GMM. In the right panel, the colors represents whether the vertices are correctly or incorrectly assigned by the *ASGE* model. The ellipses represent the 95% confidence region for the two cluster latent vectors of the estimated GMM (black) and the theoretical GMM (green). Note that misclassification occurs where the clusters are overlapping.

Figure 3.8 on the left-hand side presents a scatter plot of the embeddings, $\widehat{X}_i$, for one Monte Carlo replicate with $n = 1000$. The colors denote the true block memberships for the corresponding vertices in the SBM. The symbols denote the cluster memberships given by the GMM. In the right panel, the colors represents whether the vertices are correctly or incorrectly assigned by the *ASGE* model. The ellipses represent the 95% confidence region for the two cluster latent vectors of the estimated GMM (black) and the theoretical GMM (green). It is evident from the right-hand side of Figure 3.8 that misclassification generally occurs where the clusters are overlapping as predicted.

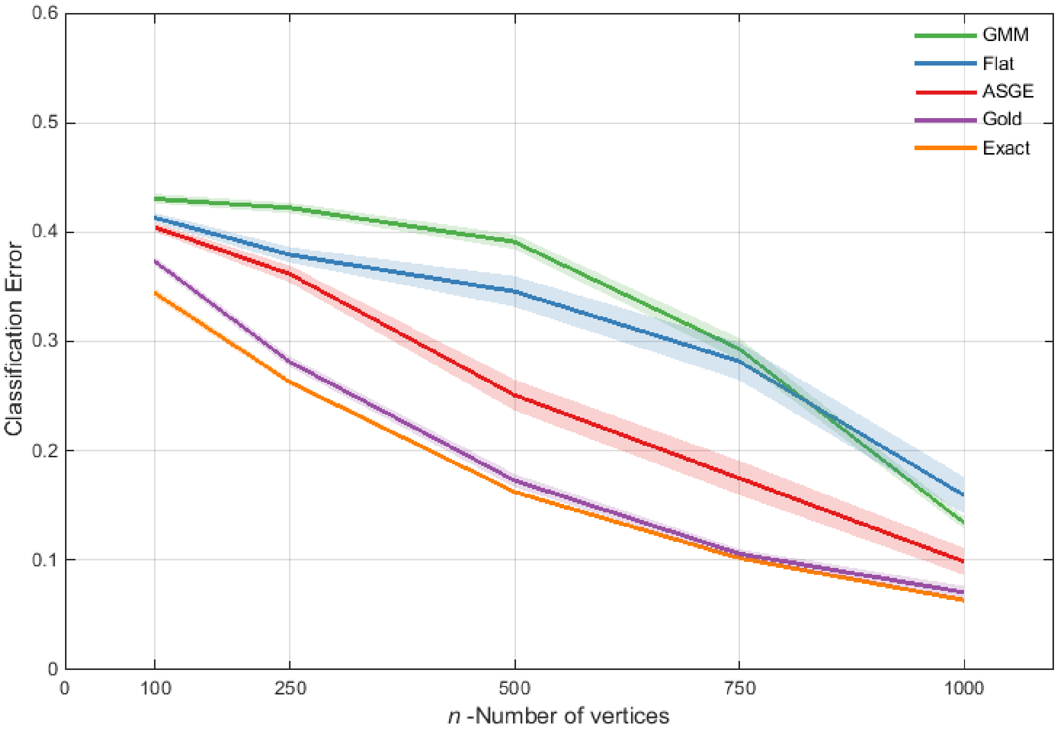


FIGURE 3.9: Comparison of vertex block assignment methodologies for the two-block SBM considered in Section 3.4.1. Shaded areas represent 95% BCA bootstrap (Efron, 1987). The plot indicates that utilizing a multivariate Gaussian mixture estimate for the embeddings as an empirical Bayes prior (*ASGE*) can yield substantial improvement over both the GMM vertex assignment and the Bayesian method with a *Flat* prior. See text for details and analysis.

Results comparing the alternative *Flat* and benchmark models are presented in Figure 3.9 and Table 3.3. As expected, the error, $\widehat{\epsilon}$, decreases for all models as the number of vertices, n, increases. As previously explained in Section 3.2, the *Exact* and *Gold* model formulated in this chapter are perceived as benchmark models; it is expected that these models will show the best performance – for the *Exact* model, all the parameters are assumed known apart from the block memberships, τ, while in the case of the *Gold* model, although ν and τ are unknown parameters, their prior distributions were taken from the true latent positions and the theoretical limiting covariances.

The main message from Figure 3.9 is that our empirical Bayes model, *ASGE*, is vastly superior to that of both the alternative *Flat* model and GMM (the sign test p-value for the paired Monte Carlo replicates is less than 10^{-10} for both comparisons for all n) and indeed nearly achieves the *Gold*/*Exact* performance by $n = 1000$.

As a side note, when we put a flat prior directly on B, we obtain results indistinguishable from our *Flat* model which places a flat prior on the latent positions, ν. In addition, instead of specifying a conjugate Dirichlet prior on the hyperparameter of the prior for τ, ρ (i.e. $\rho \mid \theta \sim \text{Dirichlet}(\theta)$ with $\theta_k = 1\ \forall k$), we also explore other choices, including:

1. using the GMM estimate of block proportions, $\widehat{\rho}$, as empirical Bayes prior for ρ,

2. using a Dirichlet distribution with θ being proportional to $\widehat{\rho}$ whose mean is $\widehat{\rho}$.

Table 3.2 summarizes the results obtained for the ASGE model using the aforementioned choices of prior on ρ. Note that the first column shows the results for the original *ASGE* model considered in this section, while the second column presents the results for option 1. As for option 2, we consider a concentration parameter, $r = \{5, 10, 50, 100, 500\}$, however we only display the results for $r = \{5, 100\}$ since no significant differences were found between $r = 100$ and the other r values. It is evident that no substantial performance gains were realized.

Table 3.2: Error rates estimates, $\widehat{\epsilon}$, with the associated 95% bootstrap confidence interval given in the parenthesis for the empirical Bayes with *ASGE* prior model based on the two-block SBM considered in this section.

	$\text{Dirichlet}(\theta_k)$	$\widehat{\rho}$	$\text{Dirichlet}(r \cdot \widehat{\rho}_k)$, $r = 5$	$\text{Dirichlet}(r \cdot \widehat{\rho}_k)$, $r = 100$
$n = 100$ 95% CI	0.4064 $[0.4022, 0.4108]$	0.4119 $[0.4069, 0.4169]$	0. 4042 $[0.4048, 0.4148]$	0.4098 $[0.4058, 0.4139]$
$n = 250$ 95% CI	0.3553 $[0.3483, 0.3624]$	0.3461 $[0.3383, 0.3539]$	0.3499 $[0.3305, 0.3692]$	0.3550 $[0.3479, 0.3620]$
$n = 500$ 95% CI	0.2307 $[0.2213, 0.2401]$	0.2388 $[0.2303, 0.2474]$	0.2205 $[0.2118, 0.2293]$	0.2345 $[0.2257, 0.2433]$

A version of Theorem 3.3 for non-dense/sparse RDPGs given in Sussman (2014), provides an empirical Bayes prior for use in non-dense regimes. Although a thorough investigation of comparative performance in this regime is beyond the scope of this research, we provide illustrative results in Figure 3.10 for the non-dense regime analogous to the setting presented in this section. For clarity, the plot only includes *ASGE* and GMM. Note that similar performance gains can also be obtained, with analogous *ASGE* superiority, in the non-dense simulation setting.

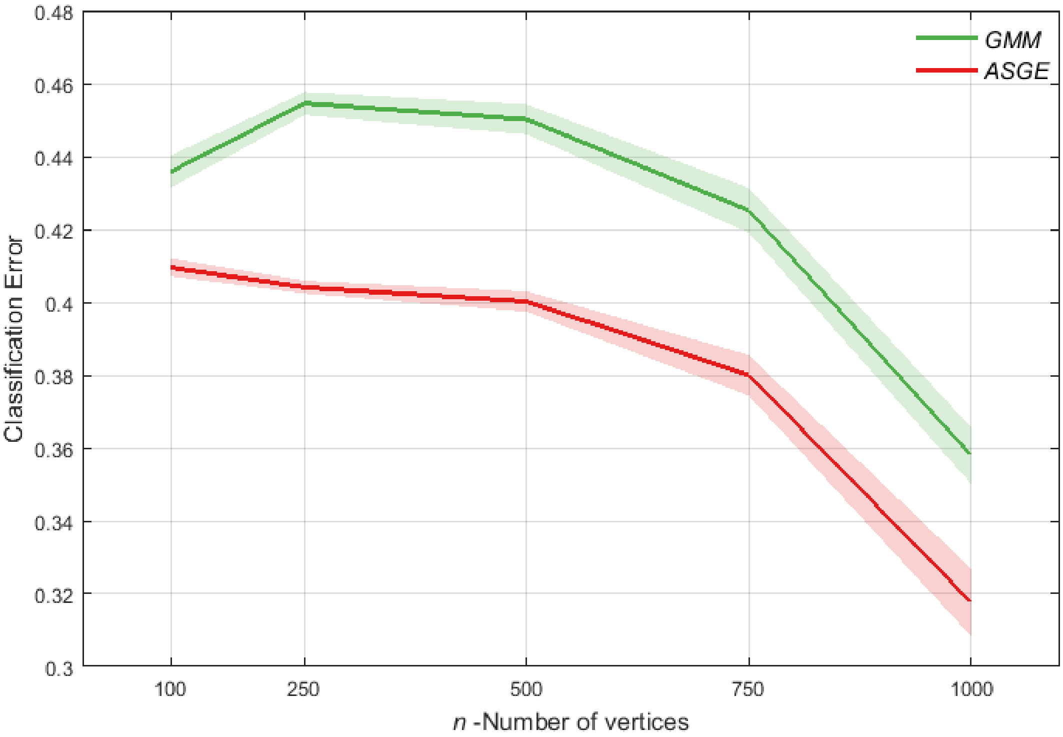


FIGURE 3.10: Illustrative results for the non-dense regime analogous to the setting considered in Section 3.4.1. Shaded areas denote standard errors. The plot suggests that we obtain similar comparative results, with analogous ASGE superiority, in a sparse simulation setting.

Table 3.3: Error rate estimates, $\widehat{\epsilon}$, with the associated 95% bootstrap confidence interval given in the parenthesis.

Model	Number of vertices n				
	100	250	500	750	1000
Exact	0.3441 $[0.3399, 0.3483]$	0.2629 $[0.2600, 0.2658]$	0.1624 $[0.1606, 0.1641]$	0.1020 $[0.1005, 0.1029]$	0.0636 $[0.0628, 0.0644]$
Gold	0.3731 $[0.3691, 0.3772]$	0.2813 $[0.2765, 0.2862]$	0.1729 $[0.1667, 0.1790]$	0.1060 $[0.1018, 0.1103]$	0.0707 $[0.0650, 0.0764]$
ASGE	0.4043 $[0.4002, 0.4084]$	0.3616 $[0.3539, 0.3693]$	0.2510 $[0.2370, 0.2650]$	0.1750 $[0.1598, 0.1910]$	0.0989 $[0.0867, 0.1112]$
Flat	0.4130 $[0.4085, 0.4176]$	0.3792 $[0.3721, 0.3862]$	0.3456 $[0.3316, 0.3596]$	0.2818 $[0.2647, 0.2989]$	0.1594 $[0.1425, 0.1762]$
GMM	0.4304 $[0.4260, 0.4348]$	0.4220 $[0.4172, 0.4268]$	0.3910 $[0.3842, 0.3978]$	0.2931 $[0.2829, 0.3032]$	0.1343 $[0.1287, 0.1398]$

3.4.2 A Dirichlet Mixture RDPG Generalization

In this section we generalize the simulation setting presented in Section 3.4.1 to the case where the latent positions are distributed according to a mixture of Dirichlets as opposed to the SBM's mixture of point masses as stated in Definition 3.1. Specifically, we consider

$$X_i \mid \nu, \rho \overset{iid}{\sim} \sum_k \rho_k \mathrm{Dirichlet}(r \cdot \nu_k).$$

Note that the SBM model presented in Section 3.4.1 is equivalent to the limit of this mixture of Dirichlets model as $r \to \infty$.

For $n = 500$, we report illustrative results using $r = 100$ with the identical number of graph realizations for comparison with the two-block SBM results from Section 3.4.1. For every realization an MCMC algorithm as previously described in Section 3.4.1.2, is run for two parallel chains until convergence is achieved. The mean error rates, $\widehat{\epsilon}$, are obtained from the last 500 iterations of the both chains. Specifically, we have the mean error rates of 0.4194, 0.2865, and 0.3705 for *Flat*, *ASGE*, and GMM respectively; the corresponding results for the SBM, from Figure 3.9, are 0.3765, 0.2510, and 0.3910. Thus we see that, while the performance is slightly degraded, our proposed empirical Bayes model works well in this RDPG generalization of Section 3.4.1's SBM. This demonstrates robustness of our method to violation of the SBM identifiability assumption. For ease of comparison, the results are summarized side-by-side below.

Table 3.4: Comparison of the error rate estimates, $\widehat{\epsilon}$, between a Dirichlet mixture RDPG generalization setting and the original setting of SBM presented in Section 3.4.1.

(a) The RDPG generalization setting

Model	$\widehat{\epsilon}$	95% bootstrap CI	Median
Flat	0.4194	$[0.4054, 0.4334]$	0.4600
ASGE	0.2865	$[0.2722, 0.3008]$	0.2580
GMM	0.3705	$[0.3576, 0.3834]$	0.3740

(b) The original setting (Section 3.4.1)

Model	$\widehat{\epsilon}$	95% bootstrap CI	Median
Flat	0.3765	$[0.3637, 0.3893]$	0.4090
ASGE	0.2510	$[0.2370, 0.2650]$	0.1800
GMM	0.3910	$[0.3842, 0.3978]$	0.4100

3.4.3 A Simulation Example with $K = 3$

In this section, we apply our models on three simulation experiments. For each simulation study, we consider the three-block SBM parameterized by

$$B(\alpha) = \alpha \begin{pmatrix} 0.3 & 0.3 & 0.3 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.3 & 0.8 \end{pmatrix} + (1-\alpha) \begin{pmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{pmatrix}, \tag{3.19}$$

where α takes the value of $\{1, 0.3, 0.13\}$ correspondingly for the three experiments. In particular, the edge probability matrix, B, in Eqn (3.19) can explicitly be expressed for each of the three experiments as

$$B(1) = \begin{pmatrix} 0.3 & 0.3 & 0.3 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.3 & 0.8 \end{pmatrix}, \quad B(0.3) = \begin{pmatrix} 0.44 & 0.44 & 0.44 \\ 0.44 & 0.50 & 0.44 \\ 0.44 & 0.44 & 0.59 \end{pmatrix}, \quad B(0.13) = \begin{pmatrix} 0.474 & 0.474 & 0.474 \\ 0.474 & 0.500 & 0.474 \\ 0.474 & 0.474 & 0.539 \end{pmatrix}.$$

In this setting manner, as α becomes closer to 0 we have an SBM with the blocks becoming harder to distinguish between them. Thus, this makes it more challenging to estimate the block memberships of vertices. Furthermore, for the respective three settings, we fix $n = 15$, $n = 150$, and $n = 300$ but with an equal number of vertices in each block (i.e. $\rho = (1/3, 1/3, 1/3)$).

Identical to Section 3.4.1, the three SBMs respectively are parametrized as RDPGs in $\mathbb{R}^3$ where the distribution of the latent positions, $\nu = [\nu_1|\nu_2|\nu_3]^\top \in \mathbb{R}^{K\times d}$ is a mixture of point masses positioned at

$$\nu_{15} = \begin{pmatrix} 0.47 & 0.17 & 0.23 \\ 0.56 & 0.41 & -0.14 \\ 0.81 & -0.38 & -0.04 \end{pmatrix}, \nu_{150} = \begin{pmatrix} 0.65 & 0.09 & 0.13 \\ 0.68 & 0.19 & -0.09 \\ 0.73 & -0.26 & -0.03 \end{pmatrix}, \nu_{300} = \begin{pmatrix} 0.68 & 0.06 & 0.08 \\ 0.69 & 0.13 & -0.06 \\ 0.71 & -0.17 & -0.02 \end{pmatrix},$$

with equal probabilities. In this section, we again assume that the dimension of the latent vectors and the number of blocks are equal and known (i.e. $d = K = 3$). For the first experiment, we independently generate 50000 SBM random graphs with values as stated in Eqn (3.19), and 1000 random graphs for both second and third experiments. To each observed graph the procedure described in Section 3.4.1 is carried out for the *ASGE* and *Flat* model. Then, for each experiment we record the mean error rates, $\widehat{\epsilon}$, together with its associated 95% bootstrap confidence interval for these models. The values are reported in Table 3.5.

TABLE 3.5: Error rate estimates for the three-block SBMs considered in Section 3.4.3.

Model	Number of vertices n		
	15	150	300
ASGE	0.5050 $[0.5024, 0.5071]$	0.5449 $[0.5400, 0.5498]$	0.6383 $[0.6363, 0.6404]$
Flat	0.5156 $[0.5135, 0.5176]$	0.5699 $[0.5650, 0.5748]$	0.6466 $[0.6448, 0.6483]$
GMM	0.5905 $[0.5888, 0.5922]$	0.6352 $[0.6300, 0.6405]$	0.6619 $[0.6595, 0.6642]$

TABLE 3.6: Sign test p-values for comparing the misassignment rates between models in Table 3.5

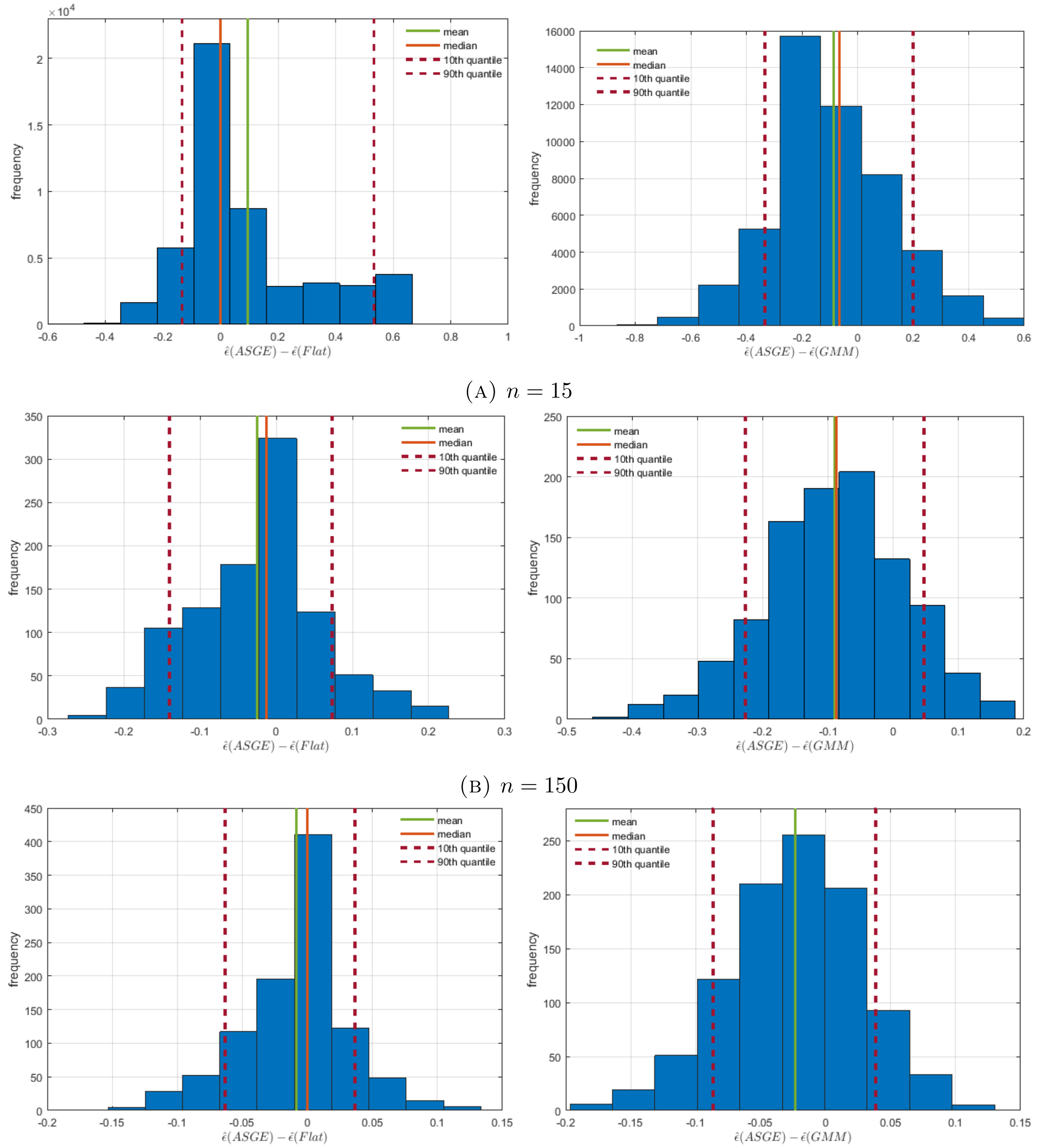
$\mathcal{H}_A$			Number of vertices n		
			15	150	300
$\epsilon(ASGE)$	$<$	$\epsilon(Flat)$	0	$< 10^{-14}$	$< 10^{-5}$
$\epsilon(ASGE)$	$<$	$\epsilon(\text{GMM})$	0	$< 10^{-83}$	$< 10^{-29}$
$\epsilon(Flat)$	$<$	$\epsilon(\text{GMM})$	0	$< 10^{-42}$	$< 10^{-17}$

Across the experiments, all models perform significantly better than chance. Notably, the error rate by chance is $2/3 = 0.667$. As α increases it becomes more difficult to stochastically distinguish between the blocks, we thus expect the mean error rate to increase, particularly going from the first experiment to the third. This prediction is evident in Table 3.5. In the first experiment, where $n = 15$, *ASGE* yields the mean error rate of 0.5050 with its 95% bootstrap CI of $(0.5024, 0.5071)$ which is still quite high due to a small graph. However, the model still comparatively outperforms the other two models in which *Flat* has the mean error rate of 0.5156 and $(0.5135, 0.5176)$ as a 95% CI, while it is 0.5905 with $(0.5888, 0.59522)$ 95% CI for GMM. These findings indeed agree with the sign test p-value for the paired Monte Carlo replicates, specifically, 0 for both *ASGE* vs *Flat* and *ASGE* vs GMM.

In the second experiment, where $n = 150$, the mean error rate for GMM is approximately 64% compared to the mean error rate for *ASGE* of approximately 55%. Based on the paired samples, the sign test p-value is less than 10^{-83} for *ASGE* vs GMM in this case. While the result of the *Flat* model appears competitive to *ASGE* in terms of mean error rate, the paired analysis shows again that the *ASGE* prior is superior as seen by sign test p-values $< 10^{-14}$. Further, considering that the mean error rates in this experiment for all models increase approximately by 4% from the first experiment *ASGE* remains the best.

Lastly, in the third experiment, where $n = 300$, all models perform poorly (near chance) as expected. Of particular note, in all three cases the *ASGE* model yields results vastly superior to both *Flat* and GMM. A summary of p-values is reported in Table 3.6. As an illustration, Figure 3.11 presents the histograms of the differential number of errors made by the *ASGE* model and *Flat* as well as *ASGE* and GMM for all three experiments. The histograms show that for most graphs, the *ASGE* model performs as well as or better than both *Flat* and GMM.

FIGURE 3.11: Histograms of the differential number of errors made by *ASGE* and Flat (*left*) together with *ASGE* and GMM (*right*) for the three-block SBMs considered in Section 3.4.3, with $n = 15$, $n = 150$, and $n = 300$, indicating the superiority of *ASGE* over GMM as well as *ASGE* over *Flat*.



(A) $n = 15$

(B) $n = 150$

(C) $n = 300$

3.4.4 Wiki Experiment

In this section we analyze an application of our methodology to the Wikipedia graph. The vertices of this graph represent Wikipedia article pages and there is an edge between two vertices if either of the associated pages hyperlink to the other. The full data set consists of 1382 vertices – the induced subgraph generated from the two-hop neighborhood of the page "Algebraic Geometry." Each vertex is categorized by hand into one of six classes – *People*, *Places*, *Dates*, *Things*, *Math*, and *Categories* – based on the content of the associated article. (The adjacency matrix and the true class labels for this data set are available at http://www.cis.jhu.edu/~parky/Data/data.html.) We analyze a subset of this data set corresponding to the three-block classes *People*, *Places*, and *Dates*, labelled here as Class 1, 2 and 3, respectively. After excluding three isolated vertices in the induced subgraph generated by these three classes, we have a connected graph with a total of $m = 828$ vertices; the class-conditional sample sizes are $m_1 = 368$, $m_2 = 269$, and $m_3 = 191$. Figure 3.12 presents one rendering of this graph (obtained via one of the standard force-directed graph layout methods, using the command `layout.drl` in the `igraph` R package); Figure 3.13 presents the adjacency matrix; Figure 3.14 presents the pairs plot for the adjacency spectral embedding of this graph into $\mathbb{R}^3$. (In all figures we use red for Class 1, green for Class 2 and blue for Class 3.) Figures 3.12, 3.13, and 3.14 indicate clearly that this Wikipedia graph violates the affinity assumption we have made for the SBM – real data will never be; nonetheless,

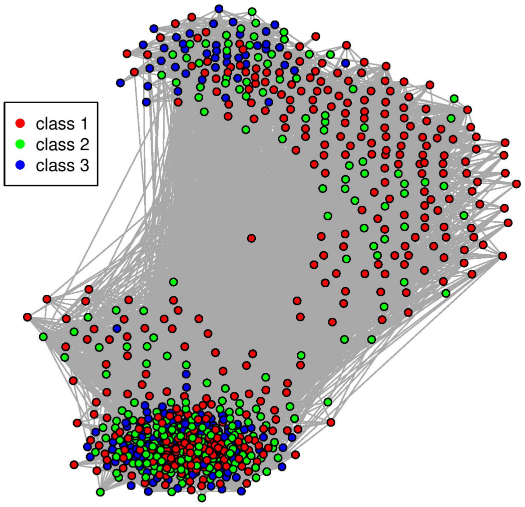


Figure 3.12: Our Wikipedia graph, with $m = 828$ vertices: $m_1 = 368$ for Class 1 = *People* = red; $m_2 = 269$ for Class 2 = *Places* = green; $m_3 = 191$ for Class 3 = *Dates* = blue.

we proceed undaunted.

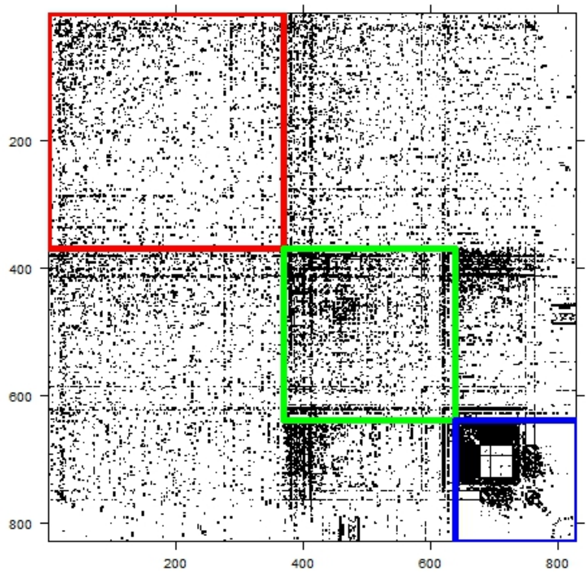


FIGURE 3.13: The adjacency matrix for our Wikipedia graph.

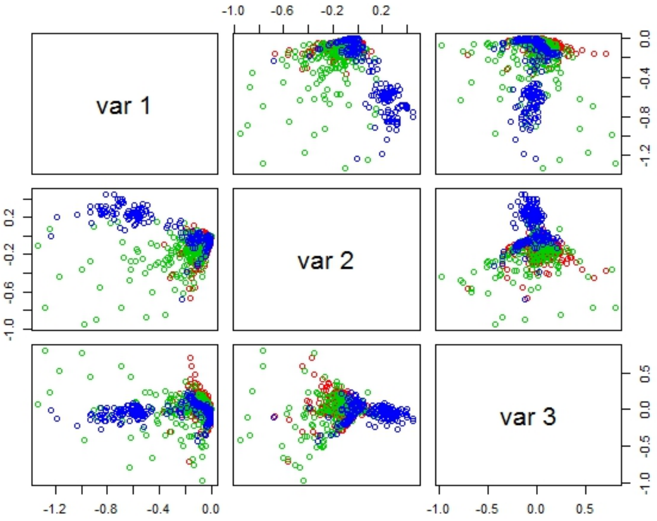


FIGURE 3.14: The adjacency spectral embedding for our Wikipedia graph.

We illustrate our empirical Bayes methodology, following Algorithm 3, via a bootstrap experiment. We generate bootstrap resamples from the embeddings of our full Wikipedia graph depicted in Figure 3.14, with $n = 300$ ($n_1 = n_2 = n_3 = 100$). This yields $\widehat{X}^{(b)}$ for each bootstrap resample, $b = 1, \ldots, 200$. As before, GMM (`MCLUST` with `EII` model in R package) is used to fit the (embedded) vertices and obtain block label estimates, $\widehat{\tau}$, mixture component means, $\widehat{\mu}_k$, and variances, $\widehat{\Sigma}_k$, for each cluster, k, of the embeddings, $\widehat{X}^{(b)}$. The clustering result from GMM for one resample is presented in Figure 3.15. We choose $d = 3$ for the embedding dimension because a common and reasonable choice is to use $d = K$; this choice is justified in the SBM case (Fishkind et al., 2013b). The GMM clustering provides the empirical prior and starting point for our Metropolis–Hasting–within–Gibbs sampling (Algorithm 4) using the subgraph of the full Wikipedia graph induced by $\widehat{X}^{(b)}$. Note that for this Wikipedia experiment, the affinity assumption on the SBM is clearly violated; as a result, we considered relaxing the constraints on the latent positions, ν, defined in Eqn (3.7) here, giving

$$\mathcal{S} = \{\nu \in \mathcal{R}^{K \times d} : \forall i, j \in [K], 0 \leq \langle \nu_i, \nu_j \rangle \leq 1\}, \tag{3.20}$$

which are utilized as hyperparameters in the *ASGE* model.

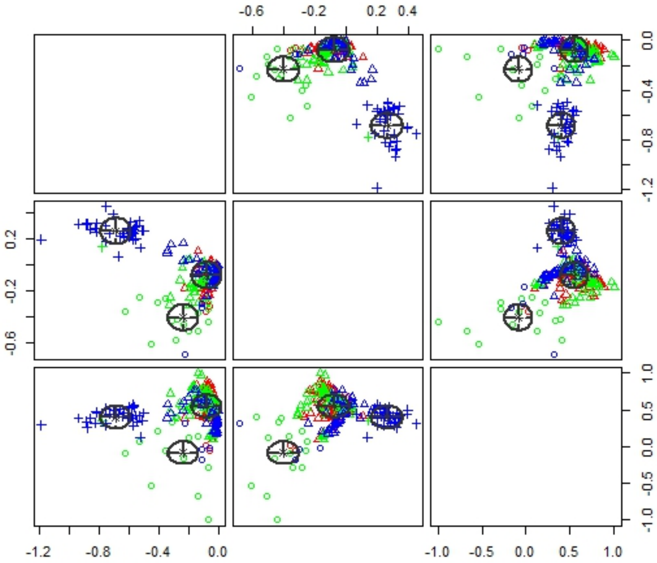


FIGURE 3.15: Illustrative empirical prior for one bootstrap resample ($n = 300$) for our Wikipedia experiment; colors represent true classes, $K = 3$ estimated Gaussians are depicted with level curves, and symbols represent GMM cluster memberships.

For each bootstrap resample, a large number of MCMC iterations for two parallel chains were simulated until convergence from the posterior densities of the block memberships, τ, and the latent positions, ν, with dispersed initial points. Identical to Section 3.4.1.2, MCMC convergence is assessed by Gelman–Rubin statistics following the procedure outlined in Algorithm 5. Figure 3.16 displays the $\widehat{R}$ of the probabilities of misassigned vertices, θ, over the last 1000 MCMC replicates for both *ASGE* and *Flat* models. It is apparent from this figure, based on the reference line at $\widehat{R} = 1.1$, that no evidence of lack of convergence was found. Figure 3.17 gives the total number of iterations until convergence of over 200 bootstrap resamples of our Wikipedia graph from both the *ASGE* and *Flat* model; Figure 3.18 displays the kernel densities fitted to posterior means for the edge probability matrix, $B = \nu\nu^\top$.

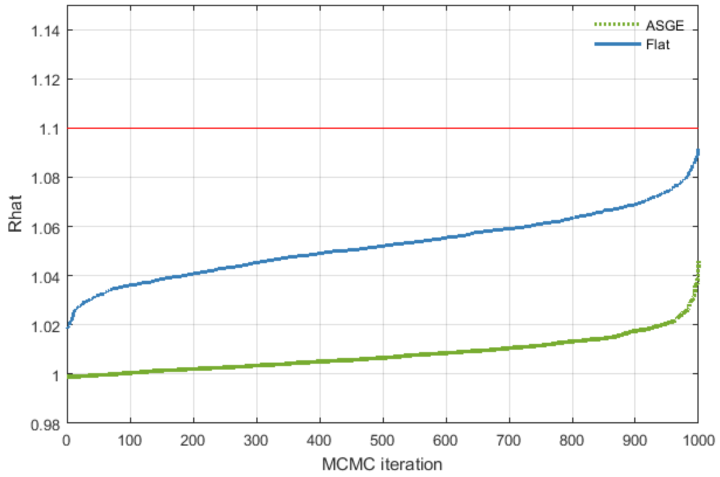
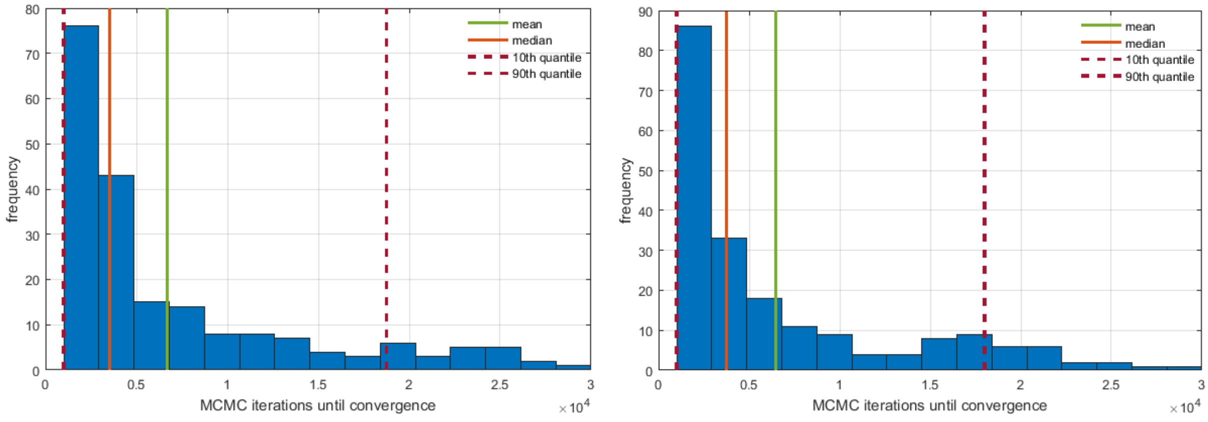


FIGURE 3.16: $\widehat{R}$ over the simulation length for the Gelman–Rubin method for convergence of the Markov chains for both *ASGE* and *Flat* models for the Wikipedia graph experiment. The red horizontal line indicates the 1.1 threshold for convergence.

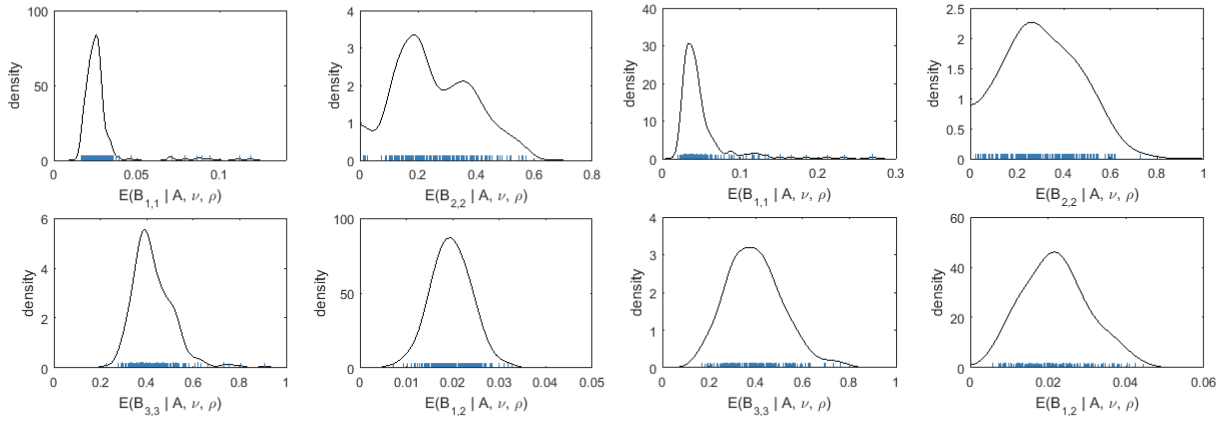


(A) *ASGE*

(B) *Flat*

FIGURE 3.17: Histograms for the total number of MCMC iterations until convergence of the empirical Bayes *ASGE* (*left*) and *Flat* model (*right*) on the Wikipedia graph experiment.

Classification results for this experiment are depicted via boxplots in Figure 3.19 and Table 3.7. We see from the boxplots that using the empirical Bayes prior does yield statistically significant improvement; indeed, our paired sample analysis yields sign test p-values less than 10^{-10} for both *ASGE* vs *Flat* and *ASGE* vs GMM (see Table 3.8). Notably, *ASGE* and *Flat* differ by 9.35% in average, which is approximately 28 different classifications per graph. Despite similar predictions, *ASGE* improves *Flat*. While GMM vs *Flat* is not statistically significant with p-values more than 0.1. Note that chance performance is $\epsilon = 2/3 = 0.67$.



(A) *ASGE* (B) *Flat*

Figure 3.18: Kernel densities fitted to posterior means of the edge probability matrix, $B = \nu\nu^\top$, over 200 bootstrap resamples of the Wikipedia graph for both *ASGE* and *Flat* models.

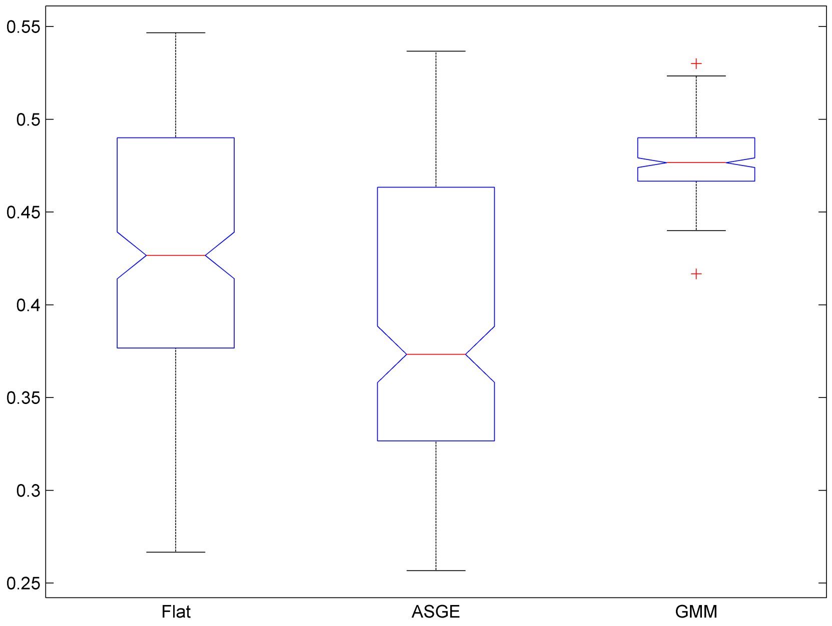


Figure 3.19: Boxplot of misassignment rates for our Wikipedia experiment.

TABLE 3.7: Estimated probability of error, $\widehat{\epsilon}$, for the three models

Model	$\widehat{\epsilon}$	95% bootstrap CI
Flat	0.4253	[0.4156,0.4350]
ASGE	0.3928	[0.3821,0.4034]
GMM	0.4774	[0.4749,0.4799]

TABLE 3.8: Sign test p-values for comparing the misassignment rates between models in Table 3.7

	$\mathcal{H}_A$		p-value
$\epsilon(ASGE)$	$<$	$\epsilon(Flat)$	$< 10^{-13}$
$\epsilon(ASGE)$	$<$	$\epsilon(\text{GMM})$	$< 10^{-13}$
$\epsilon(\text{GMM})$	$<$	$\epsilon(Flat)$	0.9170
$\epsilon(Flat)$	$<$	$\epsilon(\text{GMM})$	0.1091

3.5 Summary

In this chapter, a new empirical Bayes model for estimating block memberships of vertices in an SBM graph was proposed. Our methodology is motivated by recent theoretical advances regarding the distribution of the adjacency spectral embedding of random dot product and SBM graphs. To apply our model we derived a Metropolis-within-Gibbs algorithm for block membership and latent position posterior inference.

Our simulation experiments demonstrate that the *ASGE* model consistently outperforms the GMM clustering as well as the alternative *Flat* prior model – notably, even in our Dirichlet mixture RDPG model wherein the affinity SBM assumption is violated. Further, our simulation study also shows that the probability of misassigned vertices significantly decreases across all models as the number of vertices increases. A similar conclusion is drawn for the Wikipedia graph experiment where our *ASGE* model again performs admirably even though this real data set is far from an affinity SBM, and the embeddings of the vertices do not quite follow a mixture of Gaussians.

We considered only simple graphs; extension to directed and weighted graphs is of both theoretical and practical interest.

To avoid the model selection quagmire we have assumed throughout that the number of blocks, K, and the dimension of the latent positions, d, are known. Model selection is in general a difficult problem; however, automatic determination of both the dimension, d, for a truncated eigen-decomposition and the complexity, K, for a Gaussian mixture model estimate are important practical problems and thus have received enormous attention in both the theoretical and applied literature. For our case, Fishkind et al. (2013b) demonstrates that the SBM embedding dimension, d, can be successfully estimated, and Fraley and Raftery (2002) provides one common approach to estimating the number of Gaussian mixture components, K. We note that $d = K$ is justified for the adjacency spectral embedding dimension of an SBM, as increasing d beyond the true latent position dimension adds variance without a concomitant reduction in bias. We conducted a simulation to justify our choice of d by using twice as many dimensions used in the research (i.e. $d = 4$) which illustrates that embedding into the true model dimension, $d = 2$, is superior to $d = 4$.

In the dense regime, raw spectral embedding even without the empirical Bayes augmentation does provide strongly consistent classification and clustering (Lyzinski et al., 2013; Sussman et al., 2012a). However, this does not rule out the possibility of substantial performance gains for finite sample sizes. It is these finite sample performance gains that are the main topic of this research, and that we have demonstrated conclusively. We note that while Sussman (2014) provides a non-dense version of CLT which is briefly discussed in this chapter, both theoretical and methodological issues remain in developing its utility for generating an empirical prior. This is of considerable interest and thus a more comprehensive understanding of the CLT for non-dense RDPGs is a priority for ongoing research.

Additionally, we computed Gelman-Rubin statistics based on the percentage of misclassified vertices per iteration to check convergence of the MCMC chains. For large number of vertices, n, where perfect classification is obtainable, this diagnostic will fail; however for cases of interest (in general, and specifically in this research) in which perfect classification is beyond reasonable expectation and the empirical Bayes improves performance, this diagnostic is viable.

Finally, we note that we have made heavy use of the dot product kernel. Tang et al. (2013c) provides some useful results for the case of a latent position model with unknown kernel, but we see extending our empirical Bayes methodology to this case as a formidable challenge. Recent results on the SBM as a universal approximation to general latent position graphs (Airoldi et al., 2013; Olhede and Wolfe, 2013) suggest, however, that this challenge, once surmounted, may provide a simple consistent framework for empirical Bayes inference on general graphs.

In conclusion, adopting an empirical Bayes model for estimating block memberships in a SBM, using an empirical prior obtained from the GMM estimate for the embeddings, can significantly improve block assignment performance.

4 | VERTEX NOMINATION VIA EMPIRICAL BAYES ESTIMATION

The previous chapter considered an empirical Bayes model for community detection under the SBM graph via block membership estimation. This chapter will consider an extension of this empirical Bayes model to perform vertex nomination, a special case of the community detection task. Specifically, suppose there exists a small subset of vertices that possess attributes of interest, with only a few of these interesting vertices being observed. The vertex nomination task, proposed and further reviewed by Coppersmith and Priebe (2012) and Coppersmith (2014), aims to identify all the unobserved interesting vertices.

Vertex nomination can be formulated as a two-block SBM with one of the two blocks containing the interesting vertices. The goal is then to estimate the block memberships of vertices and subsequently construct a priority ordered "nomination list" of the vertices with unknown block memberships, such that the top of the list consists of the vertices that are in some sense the most likely to be interesting. In general, this is achieved by constructing a new likelihood model, which incorporates the additional information from the few interesting vertices whose block memberships are observed, with the previous prior specifications of the model parameters from Chapter 3. Inference about the unknown vertices can then be obtained from the resulting posterior distribution, allowing the construction of a nomination list of vertices according to their posterior probability of being in the interesting block.

This chapter begins in Section 4.1 by formally defining vertex nomination and addressing its importance along with prominent previous work. Section 4.2 introduces the vertex nomination problem and setting in the context of SBM. This is followed by a detailed description of the extended empirical Bayes method, the Metropolis-within Gibbs algorithm that implements the Bayesian inference scheme, and relevant evaluation criteria for vertex nomination in Section 4.3. Lastly Section 4.4 provides a comparison of the performance of this extended model to other alternative methods for vertex nomination on both synthetic and real–world networks.

4.1 Previous Work on Vertex Nomination

Vertex nomination, introduced by Coppersmith and Priebe (2012), is a task that aims to discover vertices that possess attributes that are of interest within a graph, and that exposing them will be of some intrinsic significance. This is often accomplished by exploiting available information about vertex and/or edge attributes in conjunction with the structure of the graph.

Consider a graph with a small subset of interesting vertices, but only a few have their identities disclosed (for example by self-disclosure or as a result of an investigation), and the remaining are unobserved. The aim is then to construct a nomination list of all unobserved vertices (i.e. interesting and uninteresting), with priority given to identifying those vertices that are most likely to be "interesting". Specifically, this task is a special example of community detection in graphs. There is a subtle distinction between classification and clustering (or supervised and unsupervised in the machine learning literature, respectively). In general, supervised and semi-supervised learnings (Bishop, 2007) consist of training sets in which objects' labels are observed (or partially observed), and they will be used to predict labels of the unknown objects. In the unsupervised learning case the machine is required to predict categories of the objects without any prior knowledge being available. Vertex nomination is analogous to the semi-supervised classification problem. In particular, the communication between all vertices (observed and unobserved labels) are utilized to infer the status of the unobserved vertices.

In many disciplines it is often of practical imperative to find vertices with some attributes of interest. Here, the term "interesting" is subjective and depends considerably on the application context. For instance, using the logging of peer-to-peer activities of individuals on child pornography networks, interesting vertices may refer to child abuse offenders, since there is evidence that convicted child pornography individuals are more likely to be child abusers. Other examples within law enforcement may be identifying potential co-conspirators in a company given a few known fraudsters, or pinpointing terrorists within the population by taking the lead from a few identified terrorists. Besides law enforcement, vertex nomination is also significant in business and social contexts; for example, finding a set of articles in relation to a specific topic, and suggesting preferences to a user via recommender systems (Resnick and Varian, 1997).

Information relevant to vertex nomination is presumed to be encoded in both the graph structure (context) and the edge attributes (content). Therefore, jointly exploiting both pieces of information may help to improve nomination performance. This brings about two broad approaches in the vertex nomination literature that hinge on the exploitation of attributed graphs (graphs which encode extra intrinsic information on vertices and/or edges; more discussion is presented in Chapter 5) or non-attributed graphs.

Firstly, operating on attributed graphs, Coppersmith and Priebe (2012) formulated the linear fusion content and context statistical model, demonstrating a superiority in nomination performance than when either approach was used alone. This fruitful outcome was also reported by Qi et al. (2012a), who used context and content links in their multimedia retrieval task. While Suwan et al. (2015) developed a Bayesian latent variable model following the work of Coppersmith and Priebe (2012) by defining context and content statistics, adapting Coppersmith and Priebe (2012)'s likelihood and introducing suitable prior distributions for the model's parameters. Inference with their model is then carried out by a Metropolis–within–Gibbs algorithm for the posterior samplings. For model comparison purposes, they mimicked the simulation setting of Coppersmith and Priebe (2012), and demonstrated that their model gives an increasingly better performance as the number of unknown interesting vertices gets smaller relative to the overall number of interesting vertices. In addition, they also showed a performance gain as the total number of interesting vertices increased. Moreover, Marchette et al. (2011) formulated an attributed version of an RDPG to integrate the additional information from edge attributes along with the edge existence. A more detailed review on these approaches will be discussed in Chapter 5.

In parallel to these developments, vertex nomination on non-attributed graphs has also been investigated by many authors including Lee et al. (2011), who proposed a multivariate self-exciting point process approach which connects the vertex memberships to the messaging activities on the topic with high risk. While Sun et al. (2012) implemented a non-parametric statistical hypothesis test approach using a Wilcoxon rank sum test based algorithm to measure the power of nominating interesting vertices given an undirected graph representation of data. To facilitate this approach, they embedded vertices into a low dimensional space by adopting the ASGE technique (Section 2.4) and multidimensional scaling (MDS) in conjunction with canonical correlation analysis (CCA). The MDS yields low dimensional latent vectors that approximately retained the dissimilarities, where the CCA maximizes the correlation for the mappings of two distinct graphs representing the same network but in a different space. They demonstrated that in both embedding approaches the fusion of embeddings from the two spaces yields a better vertex nomination power than the single space embeddings.

Recently, Fishkind et al. (2013a) developed three vertex nomination schemes based on a partially observed SBM graph, namely the canonical, graph–matching, and spectral-partitioning schemes. In terms of SBM, in order to construct a ranked list of interesting vertices the first scheme relies on the conditional probability of vertices assigned to the interesting block, given the partially observed graph realization. The drawback of this scheme is in the implementation stage where all the model parameters are assumed known priori and it is very computationally cumbersome as the number of vertices grow. Thus, this scheme is served as a comparison benchmark for other schemes. While the graph–matching scheme is based on conventional graph matching tools to rank the interesting vertices, the spectral–partitioning scheme relies on the embeddings of the adjacency matrix as previously discussed in Section 2.4; a method that is similar to Tang et al. (2013c). A more detailed review of vertex nomination can be found in Coppersmith (2014).

When the attribute of interest reflects suspicious behavior, vertex nomination can be regarded as an anomaly detection problem. Prior to the introduction of vertex nomination, the task of inferring a small region of inhomogeneity (or anomaly detection) was already attended to by many researchers in which non-attributed graphs and edge-attributed graphs were exploited. Priebe et al. (2005) proposed a theory of scan statistics on graphs which is commonly employed to examine the existence of a local signal, for example, in an image of pixel values. This technique is referred to as moving window analysis in the engineering literature. Specifically, the process includes scanning a small window over the data and computing some local statistics (e.g. average point pixel value of an image) per window, where the maximum of such locality statistics is called the scan statistic. This theory was applied to the Enron email dataset (see Section 4.5 for more information) for detecting anomaly activities in time series of graphs, and a satisfactory result was reported despite its simplicity. This anomaly detection task is further explored by Priebe et al. (2010) who studied the impact of fusion of information from both graph features and content on a time series of attributed graphs obtained from the Enron email corpus dataset. In addition, Grothendieck et al. (2010) investigated and examined how and when additional information given by attributed graphs can benefit statistical inference on random graphs, specifically, the anomaly detection task. Various anomaly detection approaches are discussed in Chandola et al. (2009).

Further, vertex nomination is distinct from, but also shares similarities with, recommender systems (Resnick and Varian, 1997). Such systems were formulated to suggest the preferences of a given recipient from the recommendations of others which received more attention after the Netflix Challenge (a competition where the user predicts film ratings only based on the information from past ratings) (Bell et al., 2008). Naturally, vertex nomination falls within a subclass of recommender systems with data encoded as attributed graphs.

As a stepping stone for the development of empirical Bayes model using edge-attributed graphs which will be investigated in Chapter 5, this chapter will first consider performing vertex nomination by exploiting a partially observed non-attributed SBM graph.

4.2 The Vertex Nomination Problem and Setting

Identical to Chapter 3, we assume a random graph, G, to be a simple undirected graph with no self–loops, multi-edges or hyper-edges with the associated adjacency matrix, $A \in \{0,1\}^{n \times n}$, being symmetric, hollow and binary.

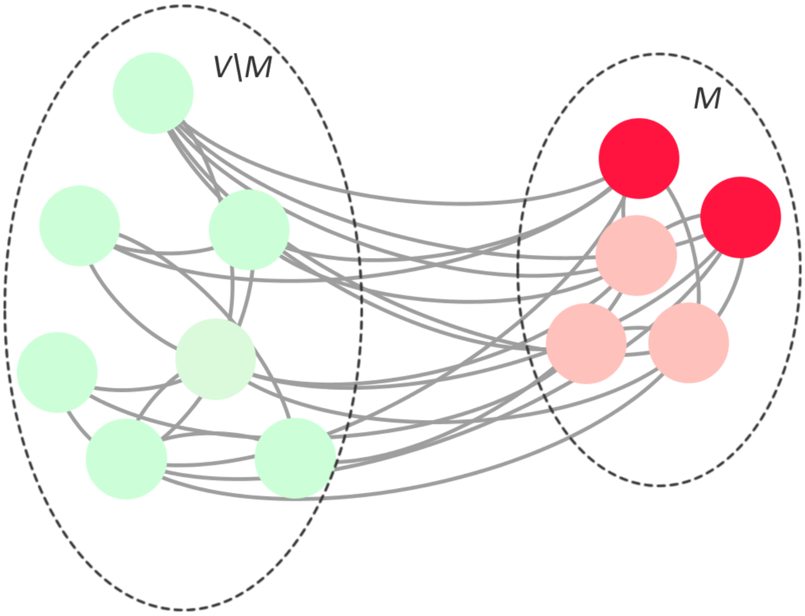


Figure 4.1: Illustrative example of the ideas underlying the model. Here, $m' = 2$ vertices are the known interesting/red vertices, $m - m' = 3$ are the unobserved interesting vertices, and $n - m = 7$ are the unobserved uninteresting/green vertices.

We formulate the vertex nomination task as a two–block SBM. Thus, given an SBM graph, this task assumes that one of the two blocks consists of interesting vertices (represented by $\mathcal{M}$) with $|\mathcal{M}| = m \ll n$; the block of uninteresting vertices is represented by $V \setminus \mathcal{M}$. Henceforth, interesting vertices are color–coded as red, while uninteresting vertices are green. A few vertices' block membership within the interesting block are disclosed which are in $\mathcal{M}'$, where $\mathcal{M}' \subset \mathcal{M}$ and $|\mathcal{M}'| = m'$. In addition, we may refer to $\mathcal{M}'$ as the 'known interesting' or observed red vertices, $\mathcal{M} \backslash \mathcal{M}'$ as the 'unobserved interesting' vertices (which we hope to nominate as interesting), and all the vertices whose block membership is unobserved as 'latent' vertices (denoted $V \setminus \mathcal{M}'$). Putting these together, we have the constraints such that $1 < m' \leq m \ll n$. Figure 4.1 demonstrates ideas underlying the model. Dark red circles signify those known interesting vertices, while light green ($V \setminus \mathcal{M}$) and red ($\mathcal{M} \setminus \mathcal{M}'$) circles signify those with the unknown block–memberships, all of which are separated by dashed borders indicating their true block memberships.

Note that the setting presented in this Chapter deliberately assumes that only a few red vertices and none of the green vertices are observed because of the intended application. Thus, each vertex in $V \setminus \mathcal{M}'$ is considered innocent unless proven guilty, as per the principle of presumption of innocence. As an example, in a fraud investigation, red vertices might represent fraudsters while green ones are non-fraudsters. Known red vertices are those proven to have committed fraud, i.e. there exists evidence of fraudulent behavior against them, whereas all of the other vertices remain latent since we cannot be certain whether or not they have committed fraud. Although, there may be situations where a few red and green vertices are known, our proposed model can potentially be adapted to accommodate such situations.

4.3 Model

This section details the empirical Bayes model; an adaptation of the original model in Chapter 3 with a focus on vertex nomination. We proceed by recapping the empirical Bayes model of Chapter 3 before discussing how the model can be extended to perform vertex nomination in Section 4.3.1. Section 4.3.2 gives details on Bayesian inference where a Metropolis–within–Gibbs algorithm is used for posterior sampling of the model parameters. In Section 4.3.3 we outline the evaluation criteria used for vertex nomination.

4.3.1 Model Description

Recall that the Athreya et al. (2015) CLT (Theorem 3.3) for RDPGs suggests that we can consider the estimated latent positions (embeddings) of a K-block SBM as (approximately) independent and identically distributed samples from a mixture of K-component multivariate normals. Chapter 3 demonstrated the utility of an estimate of this multivariate normal mixture as an empirical prior distribution for estimating block assignments of vertices in SBM graphs. The results from Chapter 3 show a substantial improvement to the block assignment performance in comparison to alternative approaches, thus providing the motivation to extend the model for vertex nomination in this Chapter.

In the setting of Definition 3.1, the probability of an edge between two vertices is assumed independent of all other edges in the graph, conditional on the block membership vector, τ, and the latent vectors, ν. The new likelihood function adapted from Eqn (3.8) can be formulated to capture the additional information from the observed interesting vertices in $\mathcal{M}'$ as follows.

$$\begin{aligned} f(A \mid \tau, \nu) = &\prod_{i=1}^{m'} \prod_{j>i}^{m'} \langle \nu_1, \nu_1 \rangle^{A_{ij}} \left(1 - \langle \nu_1, \nu_1 \rangle^{1-A_{ij}}\right) \prod_{i=1}^{m'} \prod_{j>m'}^{n} \langle \nu_1, \nu_{\tau_j} \rangle^{A_{ij}} \left(1 - \langle \nu_1, \nu_{\tau_j} \rangle^{1-A_{ij}}\right) \\ &\prod_{i=m'+1}^{n} \prod_{j>i}^{n} \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{A_{ij}} \left(1 - \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{1-A_{ij}}\right), \end{aligned} \tag{4.1}$$

where the subscript 1 in ν indicates the latent position of the interesting block.

The prior distributions for the unknown parameters - the block memberships, τ, the latent positions, ν, and the block proportion vector, ρ, are as formerly specified in Section 3.2.1. Recall that we have

$$\begin{aligned} \tau \mid \rho &\sim \text{Multinomial}(\rho), \\ \rho &\sim \text{Dirichlet}(\theta), \\ \nu \mid \widehat{\mu}, \widehat{\Sigma} &\sim \mathbb{I}_{\mathcal{S}}(\nu) \prod_{k=1}^{K} \mathcal{N}_d(\nu_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k), \end{aligned}$$

where a multinomial distribution is chosen to be a prior distribution for τ, such that ρ requires a hyperprior distribution. We place a conjugate Dirichlet distribution for ρ with θ representing the hyperparameters in the unit simplex Δ_K. The empirical Bayes prior distribution for the latent position, ν, introduced in Chapter 3 is

$$f(\nu \mid \widehat{\mu}, \widehat{\Sigma}) \propto \mathbb{I}_{\mathcal{S}}(\nu) \prod_{k=1}^{K} \mathcal{N}_d(\nu_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k). \tag{4.2}$$

where $\mathbb{I}_{\mathcal{S}}(\nu)$ as defined in Eqn (3.7) signifies the indicator function which imposes homophily and block identifiability constraints for the SBM. Additionally, $\mathcal{N}_d(\nu_k \mid \widehat{\mu}_k, \widehat{\Sigma}_k)$ indicates the density function of a multivariate Gaussian with mean, $\widehat{\mu}_k$, and covariance matrix, $\widehat{\Sigma}_k$, obtained from fitting a K-component GMM on the estimated latent positions, $\widehat{X}_1, \ldots, \widehat{X}_n$, via `MCLUST` akin to Chapter 3. However, although we observed the block memberships of a few vertices, they are assumed unobserved in the GMM fitting step. Again, $\widehat{X}_i$ is an embedding obtained from ASGE (Section 2.4).

Figure 4.2 provides a schematic view of the procedure for obtaining the empirical Bayes prior via the ASGE and GMM.

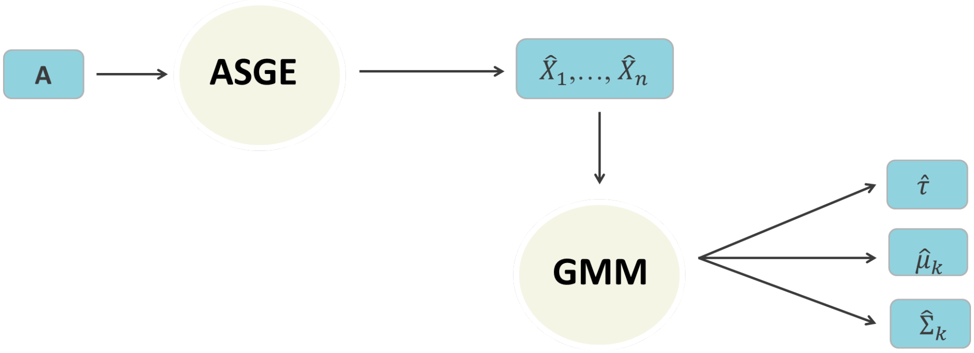


FIGURE 4.2: Schematic diagram of how to obtain the empirical Bayes prior for the latent position ν_k via ASGE and GMM. Given a graph G with adjacency matrix A, we begin by embedding each of n vertices into $\widehat{X}_i$ using ASGE, which is followed by the clustering of $\widehat{X}_i$ via GMM to obtain the clustering solutions $\widehat{\tau}$, the component means $\widehat{\mu}_k$ with the associated covariance matrices $\widehat{\Sigma}_k$.

The posterior distribution for the unknown quantities, τ, ρ, and ν, - following Bayes theorem, is given by

$$f(\tau, \nu, \rho \mid A) \propto f(A \mid \tau, \nu) \cdot f(\tau, \nu, \rho), \tag{4.3}$$

where $f(\tau, \nu, \rho)$ is the joint prior distribution. Under the assumption that τ and ν are independent of each other, we can factorize the joint density of τ, ρ, and ν, as

$$f(\tau, \nu, \rho) = f(\tau \mid \rho) \cdot f(\rho \mid \theta) \cdot f(\nu \mid \widehat{\mu}, \widehat{\Sigma}).$$

Hence, our empirical posterior can now be written as

$$f(\tau, \nu, \rho \mid A) \propto f(A \mid \tau, \nu) \cdot f(\tau \mid \rho) \cdot f(\rho \mid \theta) \cdot f(\nu \mid \widehat{\mu}, \widehat{\Sigma}).$$

In the same manner as the *ASGE* model in Section 3.2.1, we place a conjugate Dirichlet prior on ρ, and marginalize the posterior distribution over ρ to obtain the resulting posterior distribution as

$$\begin{aligned} f(\tau, \nu \mid A) &\propto f(A \mid \tau, \nu) \cdot f(\tau|\theta) \cdot f(\nu \mid \widehat{\mu}, \widehat{\Sigma}) \\ &\propto f(A \mid \tau, \nu) \cdot \left[\prod_{k=1}^{K} \Gamma(\theta_k + T_k) \right] \cdot f(\nu \mid \widehat{\mu}, \widehat{\Sigma}), \end{aligned} \tag{4.4}$$

with T_k as defined previously in Section 3.2.1. Herein, we will refer to our proposed model as the empirical Bayes stochastic blockmodel, or EBSBM for short.

4.3.2 Posterior Sampling Scheme

Similar to the posterior sampling routine described in Section 3.2.1, a Metropolis–within–Gibbs algorithm is used to sample from the posterior distribution for the EBSBM. As the components of the block label vector, τ, are binary, they can be updated sequentially via a standard Gibbs update. Let $\tau_{-i} = \tau \setminus \tau_i$ represent the block labels of vertices for all except vertex i, in which case its full–conditional distribution given τ_{-i} is

$$f(\tau_i|\tau_{-i}, A, \nu, \theta) \propto \prod_{j=m'+1, j\neq i}^{n} \langle \nu_1, \nu_{\tau_j}\rangle^{A_{ij}} (1 - \langle \nu_1, \nu_{\tau_j}\rangle)^{1-A_{ij}} \prod_{j\neq i}^{n} \langle \nu_{\tau_i}, \nu_{\tau_j}\rangle^{A_{ij}} \left(1 - \langle \nu_{\tau_i}, \nu_{\tau_j}\rangle^{1-A_{ij}}\right) \left[\prod_{k=1}^{K} \Gamma(\theta_k + T_k)\right]. \tag{4.5}$$

Clearly, the conditional posterior distribution for $\tau_i|\tau_{-i}, A, \nu, \theta \sim$ Bernoulli(ρ_i^*) where

$$\rho_{i,k}^* = \frac{f(\tau_i = k \mid \tau_{-i}, A, \nu, \theta)}{\sum_{k'=1}^{K} f(\tau_i = k' \mid \tau_{-i}, A, \nu, \theta)}, k = 1, 2. \tag{4.6}$$

A full sweep comprises of initializing τ by $\widehat{\tau}$ (the block assignment vector from fitting a two-component GMM on $\widehat{X}$), visiting each τ_i for $i = 1, \ldots, n$ and carrying out Algorithm 4 of Chapter 3, except that line 3 is now computed using Eqn (4.6). Identical to that of Section 3.2.1, an independent M-H step is used to update ν where the empirical Bayes prior, $f(\nu \mid \widehat{\mu}, \widehat{\Sigma})$, will be employed in the initialization and proposal states. See Algorithm 4 for details.

4.3.3 Performance Measures

Analogous to vertex nomination, most information retrieval and recommender systems often focus only on a few suggestions (in our case interesting vertices) as opposed to a complete classification of vertices. Thus, depending on the exploitation task at hand, evaluation measures that involve an ordered list of vertices, or where only the interesting ones are prioritized in the ordered list, are considered. In general, there are two common exploitation tasks for vertex nomination, one being identifying one vertex that is most likely to be interesting, while the other is identifying as many vertices as possible.

Let $v_{(1)}, v_{(2)}, \ldots, v_{(n-m')}$ be the ordered latent vertices in the set $V \setminus \mathcal{M}'$, where we rank vertices for nomination based on posterior probability of the vertex belonging in an interesting block.

For instance, the ranked list's beginning consists of

$$v_{(1)} = \arg\max_v \mathbb{P}(v \in \mathcal{M} \setminus \mathcal{M}'),$$

$v_{(2)}$ is the vertex which has the second largest value of $\mathbb{P}(v \in \mathcal{M} \setminus \mathcal{M}')$ and so forth. **Minimum reciprocal rank** (MRR) is the most appropriate measure if one's objective is to discover only one more interesting vertex, that is searching for the position of the first truly red vertex in the ranked list. Thus, reciprocal rank can be expressed as

$$\text{RR} = \left[\min\{i : v_{(i)} \in \mathcal{M} \setminus \mathcal{M}'\}\right]^{-1}, \tag{4.7}$$

and estimating MRR $= \mathbb{E}\,[\text{RR}]$ for a random experiment.

On the other hand, if the exploitation task is concerned with finding all the interesting vertices then **mean average precision** (MAP) is more suitable. Recall that number 1 indicates the red vertex, we define precision at rank, r, as

$$\pi(r) = \frac{\sum\limits_{i=1}^{r} \mathbb{I}_{\{1\}}(\tau_{v_{(i)}})}{r},$$

then average precision can be computed as

$$\bar{\pi} = \frac{\sum\limits_{i=1}^{n-m'} \mathbb{I}_{\{1\}}(\tau_{v_{(i)}})\pi(i)}{m - m'}. \tag{4.8}$$

Again, for a random experiment, the mean average precision is

$$\text{MAP} = \mathbb{E}[\bar{\pi}]. \tag{4.9}$$

If however, the inference task focuses on recovering k latent vertices, then the precision at rank k, $\pi(k)$, is most appropriate. More details on information retrieval tasks and other performance measures can be found in Manning et al. (2008) and Coppersmith and Priebe (2012). Note that the closer MAP and MRR are to 1 suggests a well-performed model, i.e. the better the model is at effectively prioritizing truly interesting vertices in a ranked nomination list.

4.4 Simulation Experiments

The simulation study used to demonstrate the performance of the methodology in this chapter consists of two parts. Firstly, in Section 4.4.1, we consider a two-block SBM, a small-scale simulated graph mimicking the parameter settings in Suwan et al. (2015) whose model will henceforth be abbreviated as BVN. Section 4.4.2 considers a larger two-block SBM graph adopting the parameter values in Coppersmith and Priebe (2012) (henceforth denoted C&P) and Suwan et al. (2015) for model comparison.

The MCMC Metropolis–within–Gibbs sampler of block memberships, τ, and latent positions, ν, as outlined in Section 4.3.2 are initialized at an arbitrary starting parameter vector and run for two parallel chains until convergence is achieved. This is to minimize the risk of the parameter vector being trapped in a suboptimal region. As in Chapter 3, we monitor and assess convergence of the chains by using the standard Gelman–Rubin diagnostic originating from Gelman and Rubin (1992) as formerly discussed in Section 3.4.1.2. The posterior inference for τ is then again based on the last 500 iterations of each chain after convergence, giving a combined total of 1000 posterior draws for each simulation. To quantify the nomination performance, we repeat this procedure for 1000 graph realizations and compute the performance measures, MRR and MAP, defined in Section 4.3.3.

4.4.1 Toy Example

In this section, we consider a two-block SBM parametrized by

$$B = \begin{pmatrix} 0.5 & 0.4 \\ 0.4 & 0.4 \end{pmatrix} \qquad \text{and} \qquad \rho = \left(\frac{m}{n}, \frac{n-m}{n}\right), \tag{4.10}$$

with $n = 12$, $m = 5$, and $m' = 2$. In the context of the SBM defined in Chapter 3, the block proportion vector, ρ, is essentially $(|\mathcal{M}|, |\mathcal{V} \setminus \mathcal{M}|) = \left(\frac{m}{n}, \frac{n-m}{n}\right)$ assuming that the interesting vertices are in block 1 (i.e. $\mathcal{M}$). Figure 4.3 depicts a particular graph realization with vertex 1 and 2 as the known red vertices, while 3, 4, and 5 are the latent red vertices, and the remaining vertices are in block 2 or the latent green vertices.

Recall that edge probabilities are governed by the entries in B, where the edge existence between two vertices is independent, given the vertex block memberships of the incident vertices. Again, this model can be parameterized as an RDPG in $\mathbb{R}^2$ as stated in Definition 3.1 where the distri-

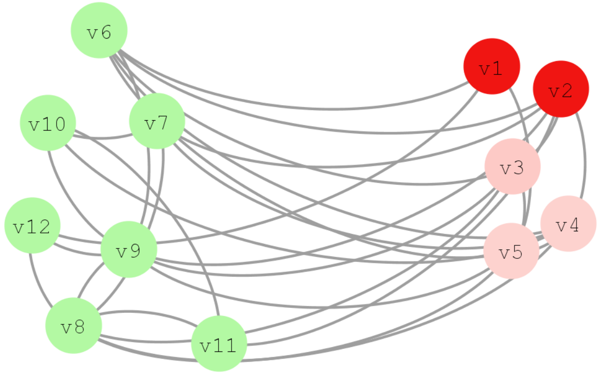


FIGURE 4.3: An example of a graph using Gephi software (Bastian et al., 2009) with 12 vertices where vertex 1 and 2 are the known red vertices, while 3, 4, and 5 are the latent red vertices, and the remaining vertices are in block 2 or the latent green vertices.

bution, F, of the latent positions is a mixture of point masses located at $\nu_1 \approx (0.5902, 0.3890)$ and $\nu_2 \approx (0.3174, 0.5470)$. For our random graphs, the vertex set is fixed but the randomness is in the edge set.

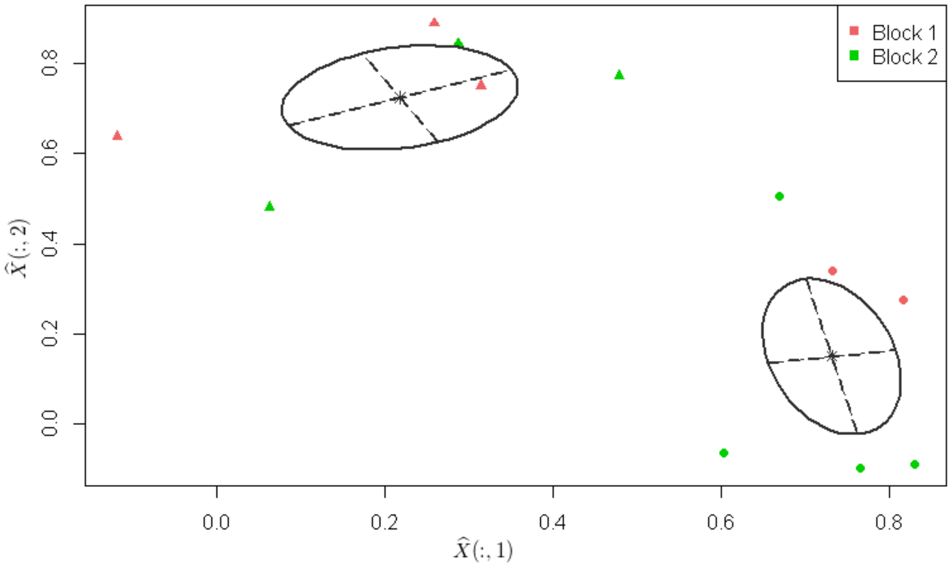


FIGURE 4.4: Clustering solutions given by the GMM procedure of $\widehat{X}_i$ for the graph in Figure 4.3. Colors indicate the true block memberships (assumed unknown when fitting), while shapes represent the clustering solutions given by the GMM. The ellipses denote the 95% confidence region for the two cluster latent vectors of the estimated GMM. The axes correspond to the $d = 2$ dimensions of the data.

Figure 4.4 depicts a scatter plot of the embeddings, $\widehat{X}_i$, for the graph example with $n = 12$ displayed in Figure 4.3. The shapes signify the clustering solutions given by the GMM and the colors signify the true block–labels, which are assumed unknown for all vertices. The ellipses denote the 95% confidence region for the two cluster latent vectors of the estimated GMM.

The concatenation of two sequences of 500 posterior samples from separate chains after convergence are used for posterior inference for the block memberships, τ. Figure 4.5 gives trace plots of the cumulative moving average estimates of the marginal posterior probabilities that each of the 10 latent vertices is red. For vertex $i \in V \setminus \mathcal{M}'$, this can be computed at iteration t by

$$\widehat{\mathbb{P}}_t(\tau_i = 1 \mid A, \tau, \nu) = \frac{1}{t}\sum_{h=1}^{t} \mathbb{I}_{\{1\}}(\tau_i^{(h)}), \tag{4.11}$$

Figure 4.5 shows that both have converged to the posterior stationary distribution; appropriate random movements at each time index with no obvious need for thinning to remove strong autocorrelation. Moreover, the estimates of the marginal posterior probabilities that each of the latent vertices is red are provided in Table 4.1. If the inference task is to identify a *single* red vertex, then it is evident from Table 4.1 and Figure 4.5 that vertex 3 will be chosen as it exhibits the highest posterior probability. In this case, it appears to be correctly identified (recall that the latent red vertices are 3, 4, and 5).

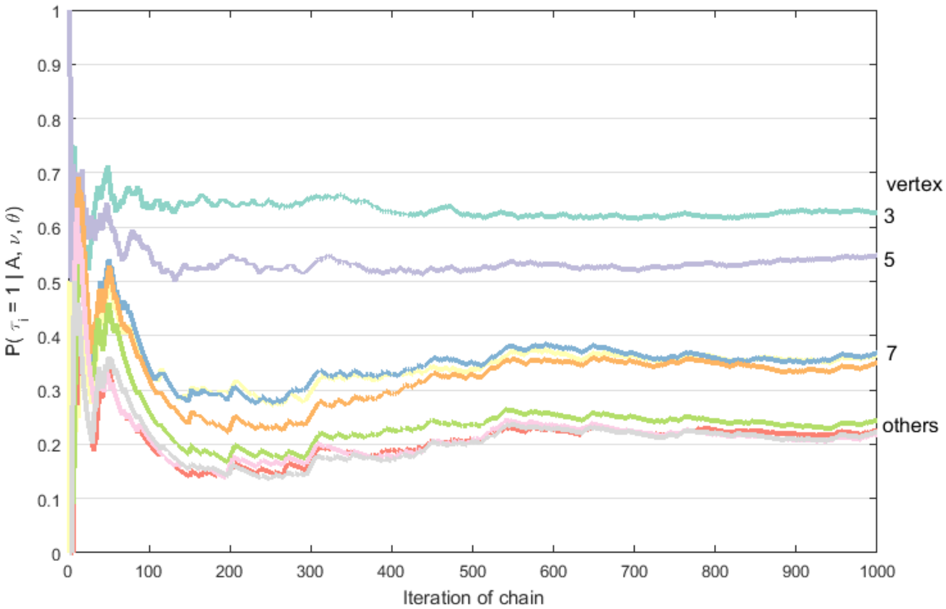


FIGURE 4.5: Trace plots of the cumulative moving average estimates of the marginal posterior probabilities that each of the latent vertices is red. The top-three ranking vertices $(3, 5, 7)$ are labeled as shown; the others $(4, 6, 8, 9, 10, 11, 12)$ are clustered together at the bottom. Recall that the three latent red vertices are 3,4 and 5, suggesting we have correctly identified the true red vertex.

One way to affirm that our MCMC chains have converged besides the Gelman Rubin diagnostic as previously described in Section 3.4.1.2 is to estimate the degree of mixing in the MCMC chain or see how well the chain is moving around the parameter space. This can be done via a visual inspection of trace plots.

Figure 4.6 shows trace plots of the edge probabilities within and between blocks 1 and 2 (i.e. $B = \nu\nu^{\top}$), while Figure 4.7 provides the cumulative moving average estimates of their marginal

TABLE 4.1: Estimates of the marginal posterior probabilities that each of the latent vertex is red (i.e. $\tau_i = 1$) for the illustrative graph with $n = 12$.

Vertex number	$\mathbb{P}(\tau_i = 1 \mid A, \nu, \rho)$
3	0.6270
4	0.2240
5	0.3640
6	0.2180
7	0.5480
8	0.3680
9	0.2430
10	0.2770
11	0.2210
12	0.3500

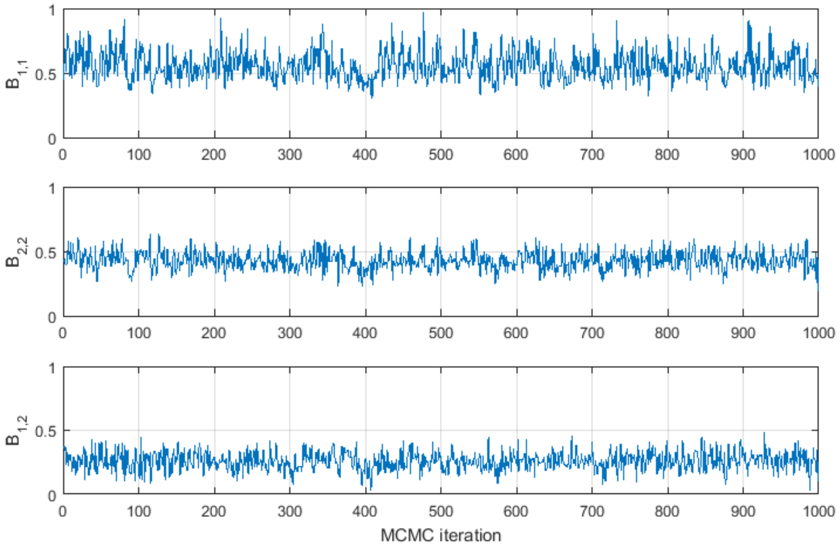


FIGURE 4.6: Trace plots of the elements in the edge probability matrix, B. Recall that the true parameter values are 0.5, 0.4, and 0.4, correspondingly.

posterior means. As an example, for $B_{1,1}$ in the first subplot of Figure 4.7, its marginal posterior mean at iteration t can be computed as

$$\widehat{\mathbb{E}}_t(B_{1,1} \mid A, \tau, \nu) = \frac{\sum_{i=1}^{t} B_{1,1}^{(i)}}{t}. \tag{4.12}$$

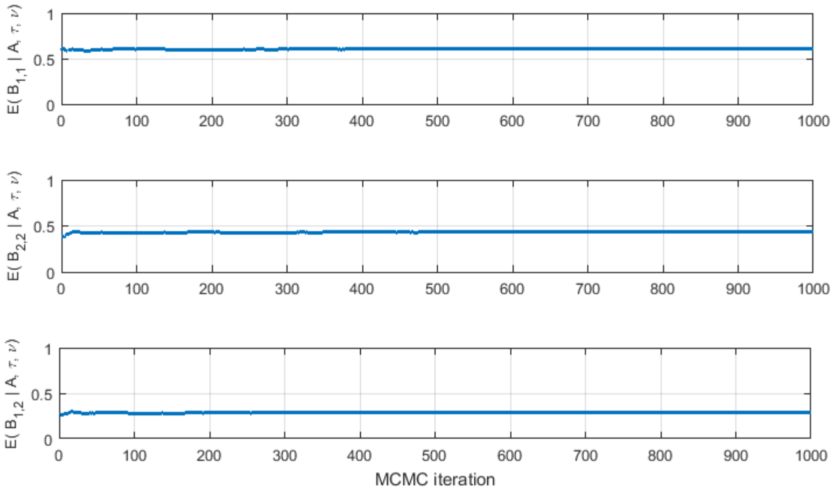


FIGURE 4.7: Trace plots of the cumulative moving average estimates of the marginal posterior means of the entries in the edge probability matrix, B.

Trace plots show clear monotonic trends, and thus give no evidence of the chains being stuck in certain areas of the parameter space that would indicate bad mixing. The posterior densities of the edge probability matrix, B, are shown in Figure 4.8. Once again, we use kernel density estimates constructed from the last 1000 MCMC iterations as in Chapter 3. Figure 4.8 illustrates the highest posterior density close to the true values of the parameters (red points).

To quantify the nomination performance, we independently duplicate the simulation for 1000 graphs obtained using the SBM parameters given in Eqn (4.10). Figure 4.9 shows the marginal distributions of the posterior means for B, clarifying that the concentration of probability density is concentrated around the true parameter values.

To aid visualization of the nomination performance across $G = 1000$ graph realizations, we also calculate sequentially down the nomination list, $j = 1, \ldots, n - m'$, the proportion of the realizations where the jth chosen vertex truly belongs in block 1. For the ordered latent vertices $v_{(j)} \in V \setminus \mathcal{M}'$ at position j, this can be computed as

$$\widehat{\mathbb{P}}_j(\tau_{v_{(j)}} = 1 \mid A, \tau, \nu) = \frac{1}{G} \sum_{g=1}^{G} \mathbb{I}_{\{1\}}(\tau_{v_{(j)}}^{g}). \tag{4.13}$$

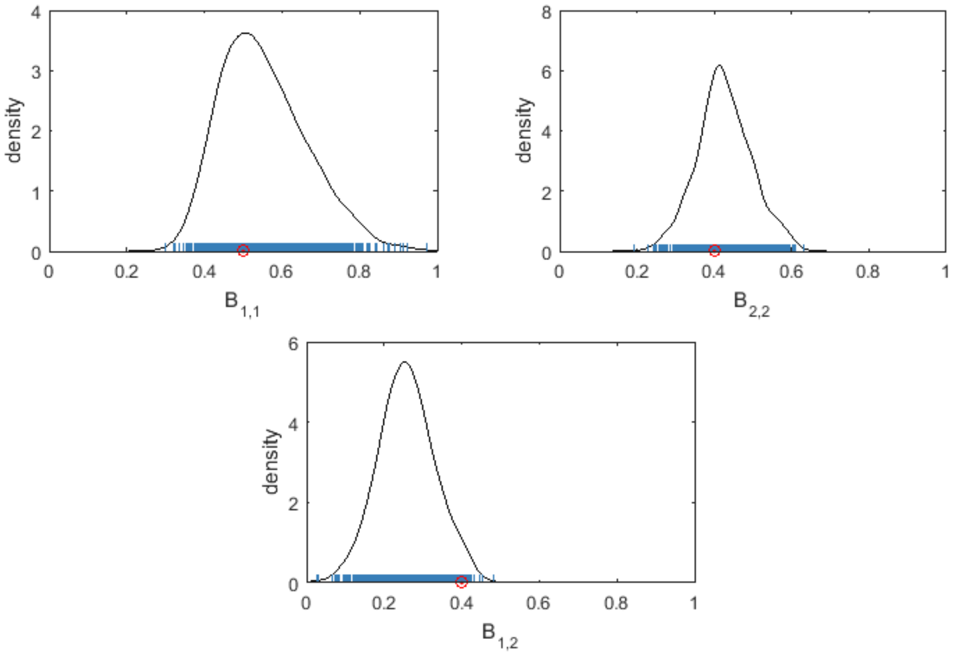


FIGURE 4.8: Posterior densities for the edge probabilities within and between blocks 1 and 2, B. Red points denote the true parameter values.

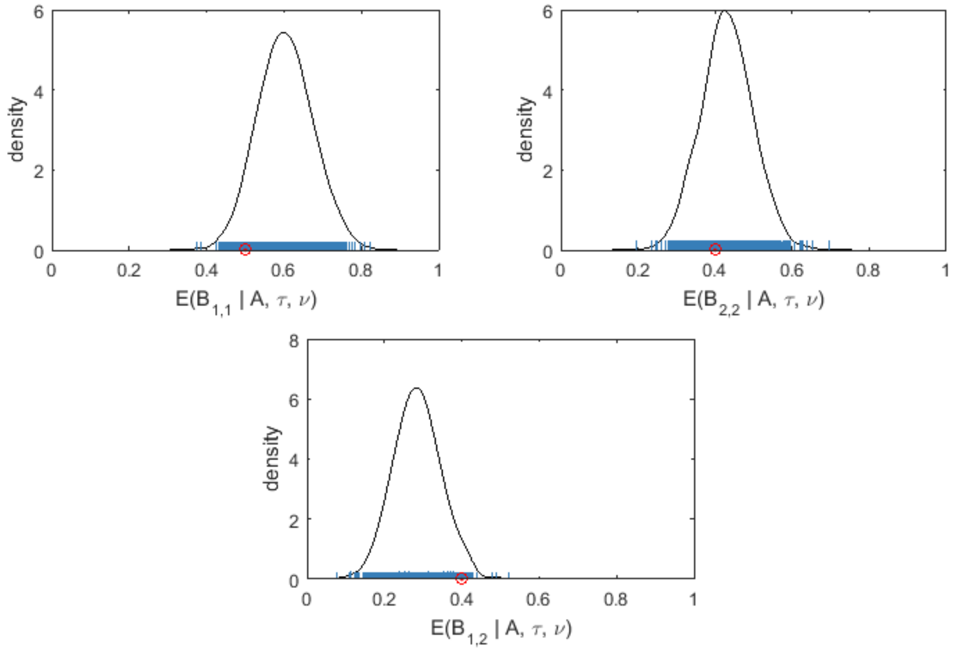


FIGURE 4.9: Kernel densities fitted to posterior means for entries of matrix B. This is obtained from 1000 graph realizations. Red points on the horizontal axis represent the true parameter values.

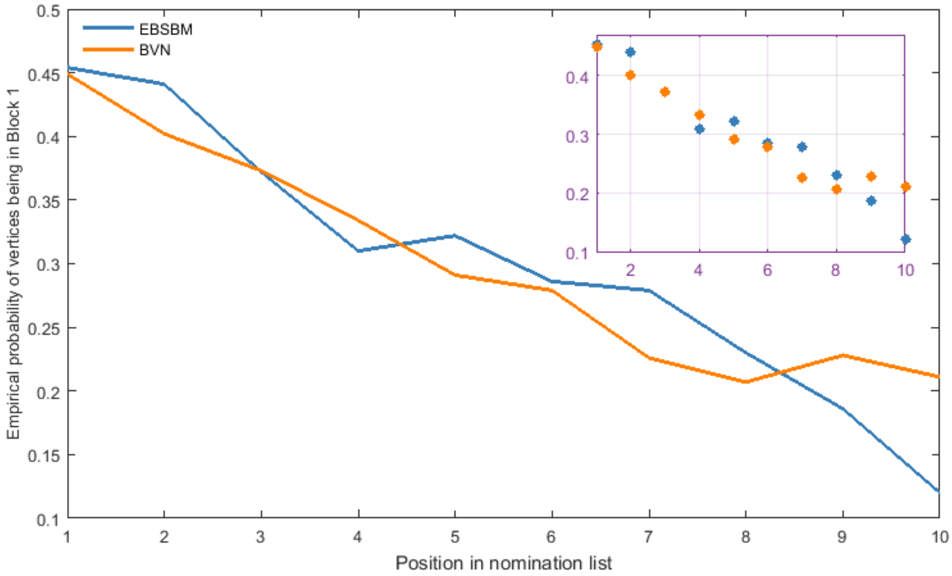


Figure 4.10: Plot of empirical probabilities of vertices which are truly in block 1 of each position descending the nomination list. The EBSBM is in blue and the BVN of Suwan et al. (2015) is in orange.

The empirical posterior probability of classifying vertices as interesting for each vertex are plotted in Figure 4.10. For comparison purposes, we include the results of BVN. It is worth noting that besides using observed edge attributes, the BVN model of Suwan et al. (2015) also leveraged latent vertex attributes by exploiting a partially observed attributed graph as part of the model's formulation. Thus, BVN is expected to have a better nomination performance than the ones operating on a simple non-attributed graph like ours. As expected, the empirical posterior probability of a vertex being classified as a member of block 1 decreases as the nomination list position increases for both models. Moreover, Figure 4.10 shows that EBSBM performs approximately as well as BVN in this case. The results also coincide with our model's MRR of 0.6641 with a 95% bootstrap confidence interval of $(0.6442, 0.6839)$, and 0.6577 along with 95% CI of $(0.6377, 0.6777)$ for BVN. Similarly, EBSBM's MAP is 0.5564 with its associated 95% CI of $(0.5436, 0.5692)$, while the MAP of BVN is 0.5413 with $(0.5290, 0.5530)$ as 95% CI. A chance performance would have $\frac{m-m'}{n-m'} = 0.3$ as its MAP. Hence, we do significantly better than chance. In addition, the sign test p-value for the paired-MCMC replicates for a significant test on MAP of EBSBM vs BVN is 0.0490.

4.4.2 Performance Comparison

To further explore the efficacy of our model, comparisons of the nomination performance are made to other competing models (i.e. BVN and C&P). This section applies the EBSBM on a larger two-block SBM graph parametrized by

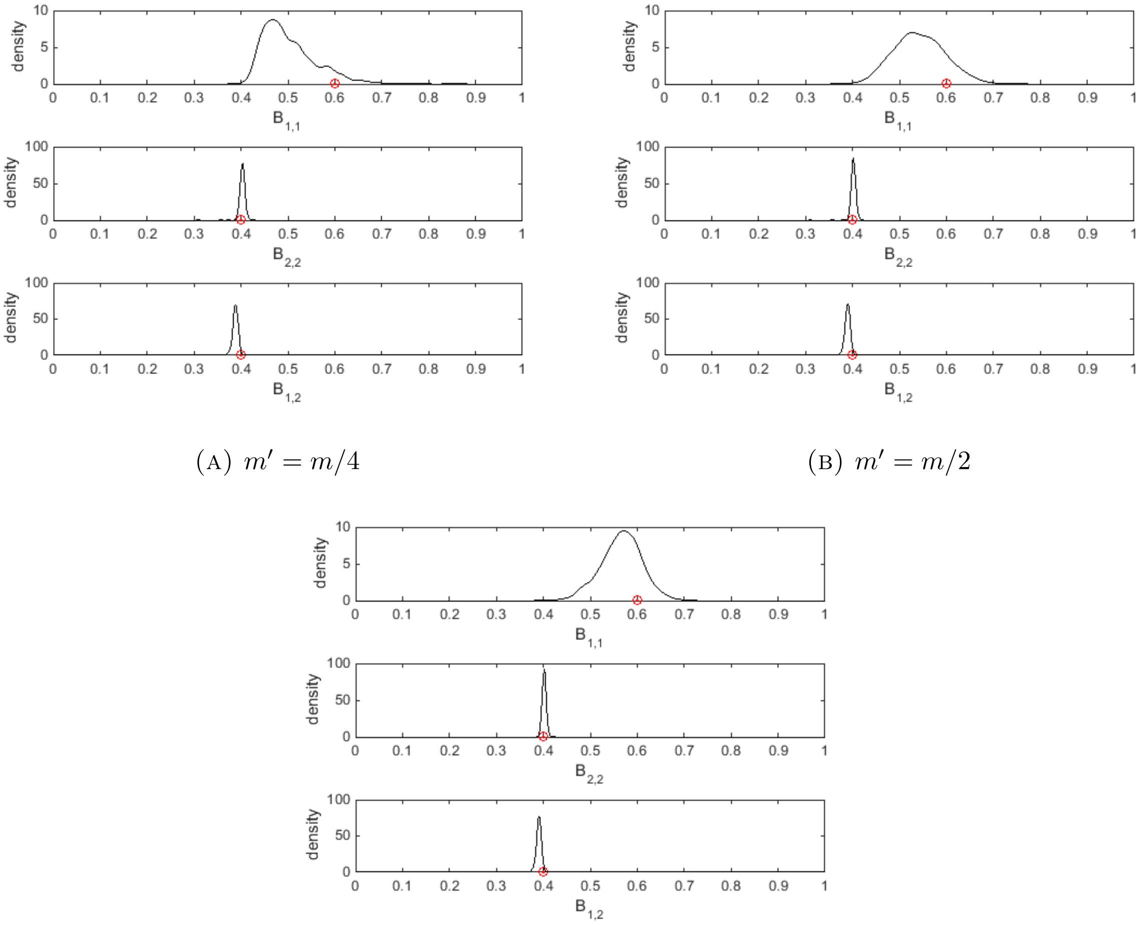
$$B = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.4 \end{pmatrix}. \tag{4.14}$$

In this experiment, we use different values for the total number of red vertices, m, and known red vertices, m', but the number of vertices, $n = 184$, is fixed. Similar to Section 4.4.1, this version of the SBM is parametrized as an RDPG centered at $\nu_1 \approx (-0.7529, 0.1822)$ and $\nu_2 \approx (-0.5878, -0.2334)$. Specifically, we look at the three sets of ratios: $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$ with $m \in \{8, 12, \ldots, 36\}$ in increments of 4. For each m of the three sets and each graph realization, we carry out the algorithms described in Section 4.3.2 for posterior sampling and record the nomination performance via the evaluation criteria, MAP and MRR, stated in Section 4.3.3. This procedure is repeated 1000 times and the results are compared with BVN and C&P.

Figure 4.11 displays the marginal distributions of the posterior means of the entries of B obtained over 1000 graph replicates, where $m = 24$ for all three sets as an illustration. These figures indicate highest density estimates of $B_{1,2}$ and $B_{2,2}$ near the true values, but less so for $B_{1,1}$ seemingly due to a small m. However, the bias of $B_{1,1}$ reduces as m increases.

With the same setting as the previous figure (i.e. $m = 24$), we can visualize the nomination performance of EBSBM against BVN via the plots of the empirical probabilities of vertices that are truly red for each position as we go down the nomination list, for all three sets as displayed in Figure 4.12. Generally, the two models appear to be very competitive. However, if we look closely at the empirical probabilities of the first positioned vertex being a true member of the interesting block, then it is clear that EBSBM achieves comparatively higher probabilities of vertices being red around the beginning of the nomination list. This indicates that EBSBM yields a better nomination performance than BVN, particularly in the first two subplots (i.e. $m' = m/4$ and $m/2$). However, as the number known red vertices increases (i.e. m' increases), BVN gives comparable results to EBSBM.

For an overall nomination performance of EBSBM in this simulation study, Figure 4.13 presents plots of the two evaluation measures described in Section 4.3.3, comparing various competing



(A) $m' = m/4$

(B) $m' = m/2$

(C) $m' = 3m/4$

Figure 4.11: Examples of kernel densities fitted to posterior means of the components in B obtained from 1000 graphs for all three simulation sets, $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$, where $m = 24$.

models (i.e. BVN and C&P) across all three sets of ratios as we increase m'. Note that as we only have the approximate MAP values of C&P, 95% bootstrap CIs of MAP are incalculable and thus are not displayed. Figure 4.13 shows undoubtedly that all models perform significantly better than chance throughout the three simulation sets. When the value of m is fixed moving from set to set (i.e. $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$), that is having an increase in the number of known red vertices, m', all models show performance improvements across sets as predicted. In addition, it is evident that the MAP value increases for all models indicating gradual performance gains as the total number of red vertices, m, (and thus m') increases. However, for small values of m, these models perform equally poorly, for instance when $m = 8$, MAP are all below 0.25. When $m' = m/4$ (top right figure), C&P gives the best performance with small m followed by BVN and EBSBM. As m increases, particularly when it is greater than 24, our method continuously performs better compared to the alternative methods. This visual inspection coincides with our paired sample analysis using the sign-test which yields p-values less than 10^{-5} for EBSBM vs BVN. Next, when

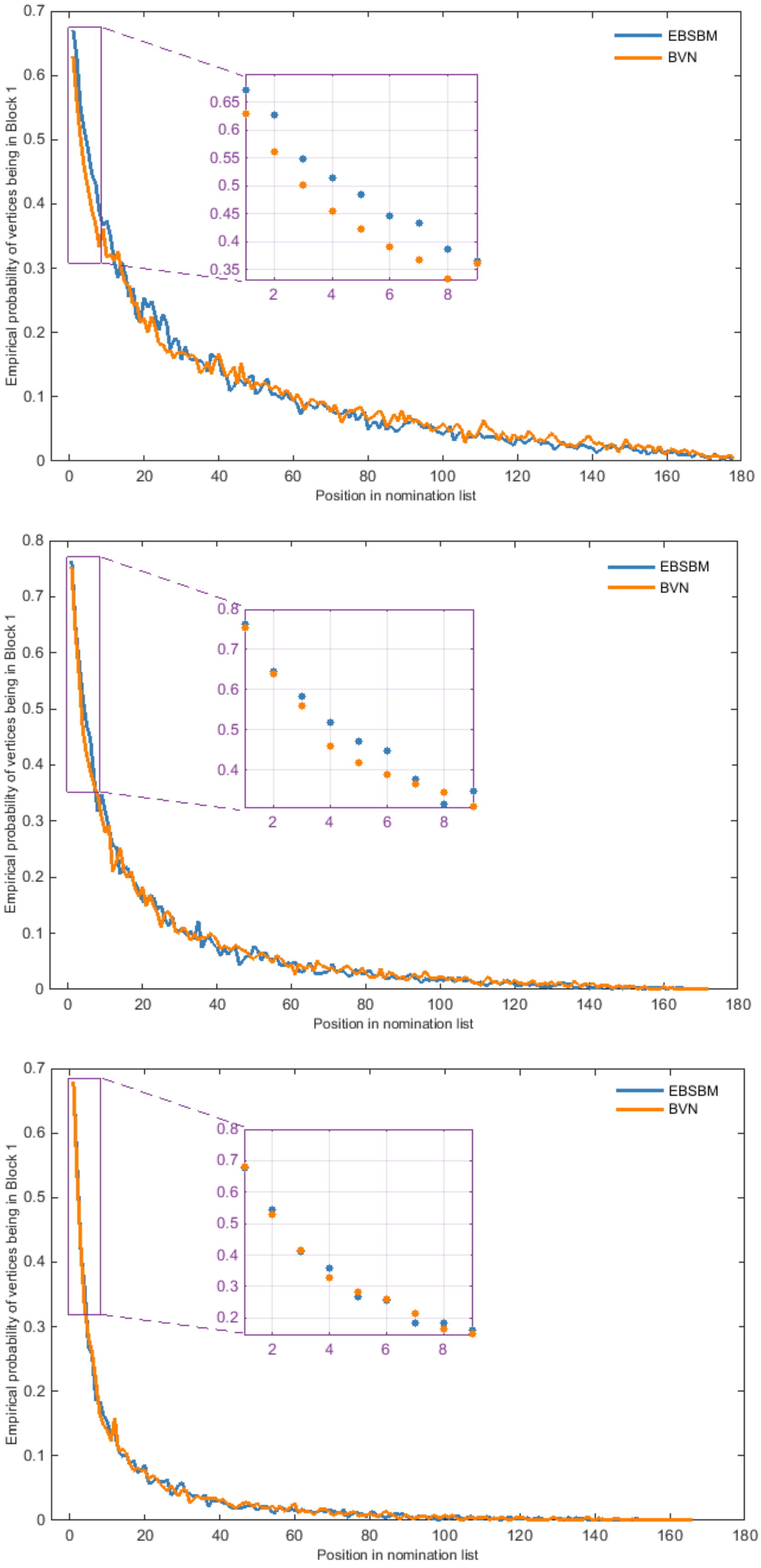


FIGURE 4.12: Comparison of empirical probabilities of vertices being a member of the block of interest against position in nomination list obtained from 1000 graphs, where $n = 184, m = 24$ for all three sets for EBSBM (blue) and BVN (orange). Columns, from left to right, represent $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$, where $m = 24$.

$m' = m/2$ (middle right figure), MAP values of all models start off roughly the same and at $m = 20$ and beyond, EBSBM shows a comparatively excellent performance while BVN and C&P perform at a similar level throughout. In the last set of simulations (i.e. $m' = 3m/4$), these models are comparable across m values. Despite the fact that EBSBM excludes the additional edge attribute information, unlike BVN and C&P, we still eminently perform as well as or better than theirs.

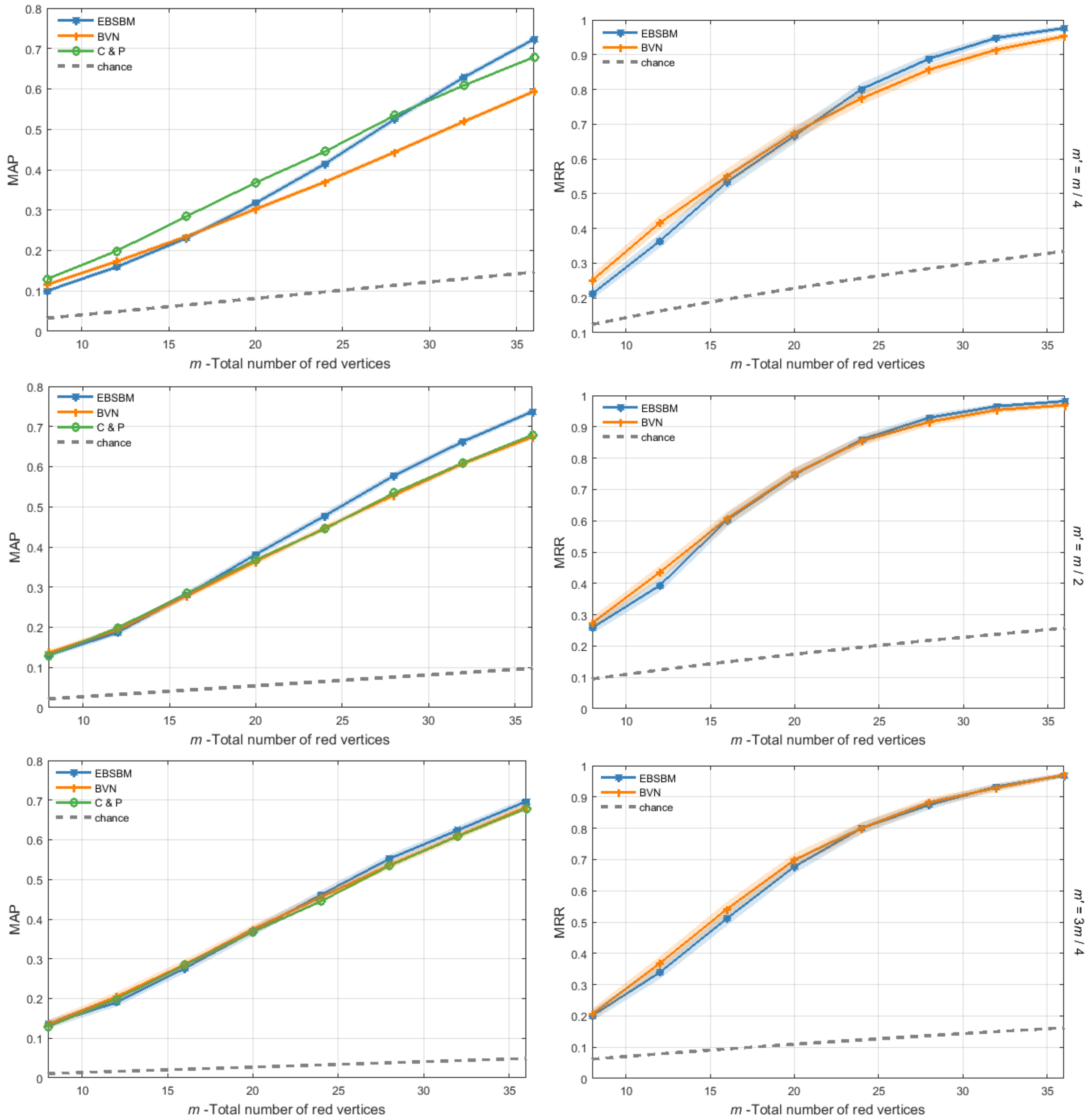


FIGURE 4.13: The nomination performance according to mean average precision (MAP) and minimum reciprocal rank (MRR) (*left* and *right*, respectively) across three sets of ratios. Rows, from top to bottom, indicate $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$. The results are obtained from 1000 graph realizations. The x-axis represents increasing values of m and the y-axis represents MAP and MRR values, correspondingly. Shaded areas in EBSBM and BVN models represent 95% bootstrap standard errors.

When the exploitation task is to identify from the latent vertices only one red vertex, apart from visual inspection of the empirical probabilities of vertices that are truly red (an example given in Figure 4.12), the MRR value should be used to evaluate the performance (see the second column plots of Figure 4.13). In the case of MRR, similar remarks can be made. These include: 1) these MRR values are exceptionally greater than the MRR for chance, 2) when fixing m, MRR increases from set to set (i.e. top to bottom), and 3) as m increases MRR also progressively approaches 1 for both EBSBM and BVN models. More precisely, for $m' = m/4$, EBSBM starts obtaining a better performance than BVN when $m > 24$ and $m > 28$ in the case of $m' = m/2$. Analogous to MAP, both models give a similar performance when $m' = 3m/4$. A summary of MAP and MRR values of both EBSBM and BVN, together with their associated 95% bootstrap CI are reported in Table 4.2

Table 4.2: MAP and MRR with the associated 95% confidence interval given in the parenthesis.

Model		$m' = m/4$							
		8	12	16	20	24	28	32	36
EBSBM	MAP	0.0998	0.1593	0.2303	0.3174	0.4152	0.5254	0.6296	0.7233
	95% CI	$[0.0950, 0.1050]$	$[0.1530, 0.1650]$	$[0.2240, 0.2370]$	$[0.3100, 0.3250]$	$[0.4070, 0.4230]$	$[0.5170, 0.5340]$	$[0.6220, 0.6380]$	$[0.7160, 0.7300]$
	MRR	0.2120	0.3630	0.5340	0.6660	0.8010	0.8880	0.9480	0.9760
	95% CI	$[0.1960, 0.2290]$	$[0.3430, 0.3840]$	$[0.5110, 0.5560]$	$[0.6450, 0.6880]$	$[0.7830, 0.8190]$	$[0.8740, 0.9030]$	$[0.9380, 0.9580]$	$[0.9690, 0.9830]$
BVN	MAP	0.1150	0.1730	0.2350	0.3030	0.3700	0.4440	0.5200	0.5940
	95% CI	$[0.1100, 0.1200]$	$[0.1680, 0.1780]$	$[0.2290, 0.2400]$	$[0.2960, 0.3090]$	$[0.3640, 0.3760]$	$[0.4380, 0.4500]$	$[0.5140, 0.5260]$	$[0.5880, 0.6000]$
	MRR	0.2500	0.4160	0.5500	0.6750	0.7740	0.8570	0.9140	0.9530
	95% CI	$[0.23300.2690]$	$[0.3940, 0.4370]$	$[0.5280, 0.5720]$	$[0.6540, 0.6970]$	$[0.7550, 0.7930]$	$[0.8410, 0.8720]$	$[0.9020, 0.9270]$	$[0.9430, 0.9620]$
		$m' = m/2$							
EBSBM	MAP	0.1310	0.1870	0.2810	0.3810	0.4780	0.5780	0.6630	0.7370
	95% CI	$[0.1240, 0.1380]$	$[0.1790, 0.1950]$	$[0.2720, 0.2900]$	$[0.3720, 0.3910]$	$[0.4680, 0.4880]$	$[0.5690, 0.5870]$	$[0.6550, 0.6710]$	$[0.7300, 0.7440]$
	MRR	0.2580	0.3940	0.6030	0.7480	0.8600	0.9300	0.9660	0.9820
	95% CI	$[0.2400, 0.2770]$	$[0.3720, 0.4150]$	$[0.5810, 0.6260]$	$[0.7280, 0.7680]$	$[0.8440, 0.8760]$	$[0.9170, 0.9410]$	$[0.9570, 0.9740]$	$[0.9760, 0.9870]$
BVN	MAP	0.1370	0.1940	0.2780	0.3640	0.4480	0.5290	0.6080	0.6740
	95% CI	$[0.1300, 0.1440]$	$[0.1870, 0.2010]$	$[0.2700, 0.2860]$	$[0.3550, 0.3720]$	$[0.4400, 0.4550]$	$[0.5220, 0.5370]$	$[0.6010, 0.6140]$	$[0.6670, 0.6800]$
	MRR	0.2740	0.4350	0.6080	0.7510	0.8560	0.9160	0.9540	0.9690
	95% CI	$[0.2550, 0.2930]$	$[0.4130, 0.4570]$	$[0.5860, 0.6300]$	$[0.7310, 0.7700]$	$[0.8390, 0.8710]$	$[0.9030, 0.9280]$	$[0.9440, 0.9630]$	$[0.9610, 0.9760]$
		$m' = 3m/4$							
EBSBM	MAP	0.1350	0.1900	0.2760	0.3690	0.4620	0.5520	0.6240	0.6970
	95% CI	$[0.1240, 0.1460]$	$[0.1790, 0.2010]$	$[0.2640, 0.2870]$	$[0.3570, 0.3810]$	$[0.4500, 0.4730]$	$0.5420, 0.5630$	$[0.6140, 0.6350]$	$0.6880, 0.7060$
	MRR	0.2010	0.3390	0.5120	0.6770	0.8000	0.8750	0.9330	0.9690
	95% CI	$[0.1830, 0.2190]$	$[0.3170, 0.3600]$	$[0.4890, 0.5350]$	$[0.6550, 0.6990]$	$[0.7820, 0.8190]$	$[0.8590, 0.8900]$	$[0.9210, 0.9440]$	$[0.9600, 0.9760]$
BVN	MAP	0.1350	0.2050	0.2860	0.3750	0.4550	0.5370	0.6100	0.6830
	95% CI	$[0.1260, 0.1460]$	$[0.1940, 0.2160]$	$[0.2740, 0.2970]$	$[0.3640, 0.3860]$	$[0.4450, 0.4660]$	$[0.5270, 0.5470]$	$[0.6000, 0.6200]$	$[0.6740, 0.6910]$
	MRR	0.2060	0.3680	0.5420	0.6990	0.8010	0.8830	0.9290	0.9700
	95% CI	$[0.1890, 0.2240]$	$[0.3460, 0.3900]$	$[0.5180, 0.5650]$	$[0.6770, 0.7200]$	$[0.7820, 0.8190]$	$[0.8680, 0.8970]$	$[0.9170, 0.9400]$	$[0.9620, 0.9780]$

4.5 Application to Enron

The Enron Corporation was a large energy and trading company specializing in the marketing of natural gas, electricity, and communications. In the year 2000, it was reported to be the seventh largest company in the United States, with claimed revenues of over \$100 billion. However, Enron abruptly went bankrupt in the following year due to internal accounting fraud committed by top executives and associates. Subsequently, the Enron email corpus data has been employed by many researchers for various tasks including visualization, document classification, and social network analysis. The data comprised of email communications which included fraudulent activity among Enron associates and senior management. The data is available online at `http://www.enron-mail.com`. For our analysis, the vertices of this Enron graph indicate Enron associates and senior management. There is an edge between two vertices if there is at least one message between them.

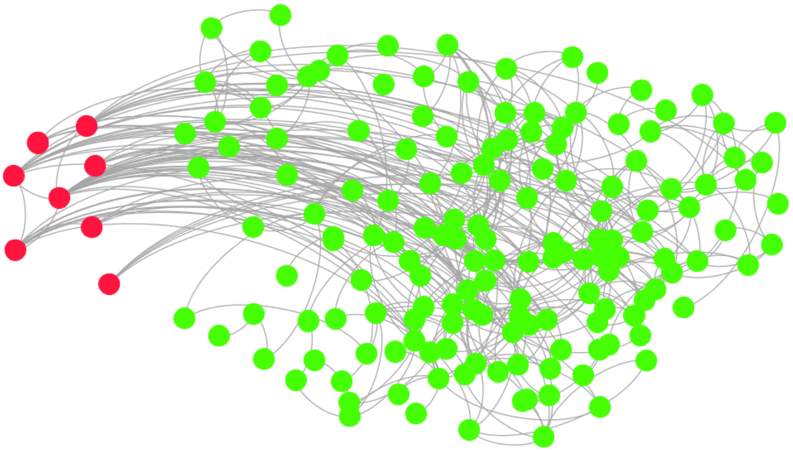


Figure 4.14: Our Enron graph after excluding isolated vertices, with $n = 184$ vertices: $m = 10$ for Class 1 = red = fraudsters, $n - m = 174$ for Class 2 = green = non-fraudsters.

Priebe et al. (2005) processed the data and focused on a period from 1998 to 2002 resulting in 189 graphs (1 graph per week). Each graph consists of $n = 184$ distinct email users, 10 of which are ascertained to have committed financial fraud. We restrict our attention to week 38 of Priebe et al. (2005)'s Enron graph which corresponds to the $K = 2$ classes *fraudsters* and *non-fraudsters*, which can be divided into $\mathcal{M}$ and $V \setminus \mathcal{M}$ where $\mathcal{M}$ represents a small set of vertices ($m = 10$) communicating at a higher rate compared to $V \setminus \mathcal{M}$. We also label these classes as red/Class 1 and green/Class 2, respectively.

For our experiment, we consider 5 of the 10 fraudsters as known red vertices and the remaining vertices as unobserved. We illustrate our methodology on 252 graphs taken from combinations

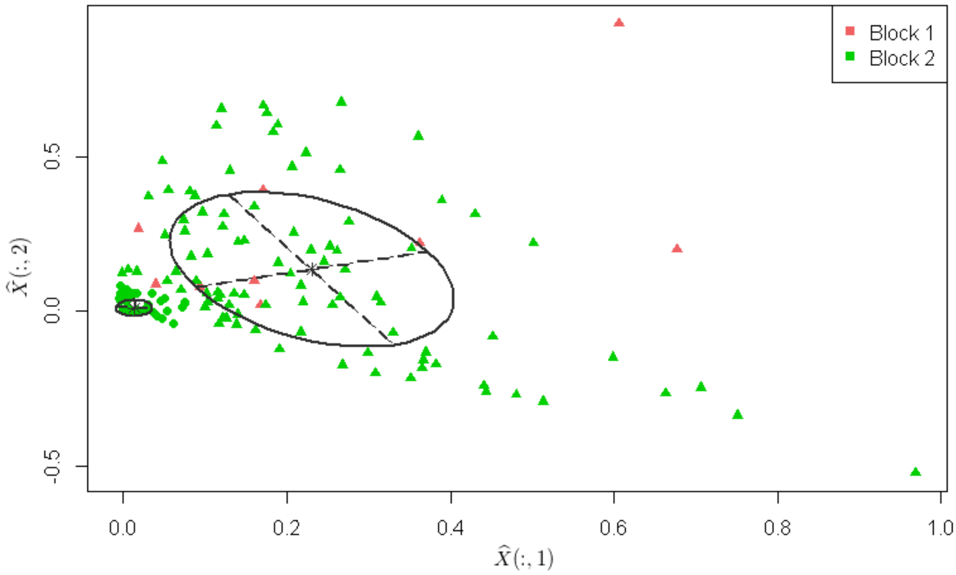


FIGURE 4.15: The adjacency spectral embedding for our Enron graph with colors signifying vertices' true class, and symbols signifying the GMM clustering solutions. The ellipses signify the 95% confidence region for the two cluster latent vectors of the estimated GMM.

of 5 known red vertices from the 10 fraudsters (i.e. 10 choose 5). The principle aim of the study is therefore to evaluate the performance of our methodology to identify the other 5 fraudsters that have been assumed to have an unknown status. Figure 4.14 shows one rendering of this graph obtained by the software `Gephi` (Bastian et al., 2009). Figure 4.15 presents the adjacency spectral embedding of this graph into $\mathbb{R}^2$ with colors indicating their true class, and symbols indicating the clustering solutions given by the GMM. We can see clearly from this figure that the embeddings of the vertices for the Enron graph do not quite follow a mixture of Gaussians; however, we will proceed with the analysis regardless.

Following the procedure described in Section 4.3, the distributions of the marginal posterior means of the components in B from the 252 combinations are given in Figure 4.16. The plots display approximately higher posterior means for edge probabilities in the interesting block than the uninteresting block, and exceptionally higher than posterior means for edge probabilities between the two blocks. The sample means over all 252 combinations of the posterior means edge probability parameters are $\bar{B}_{1,1} = 0.0950$, $\bar{B}_{2,2} = 0.0138$, and $\bar{B}_{1,2} = 0.0094$.

For visual inspection of the nomination performance we can observe the empirical probabilities of vertices truly being in block 1 against the nomination position via Figure 4.17. Our model evidently yields substantially higher empirical probabilities of vertices being classified as fraudsters than BVN's in the first few positions, indicating a better nomination performance. Immediately after, these probabilities in both models rapidly drop down to nearly 0. The highest empirical

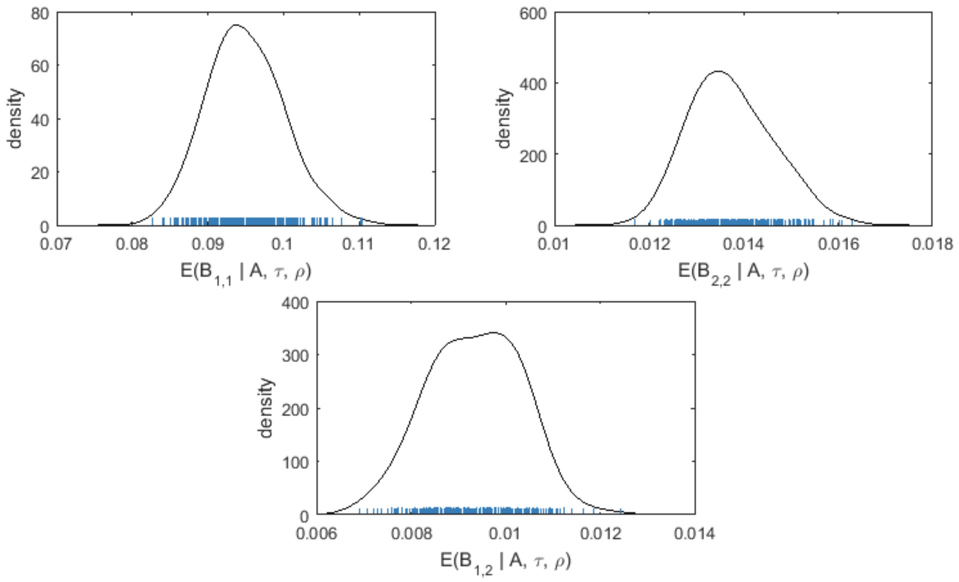


FIGURE 4.16: Kernel densities estimates for the posterior means for the edge probability matrix,B, obtained from the 252 graph combinations.

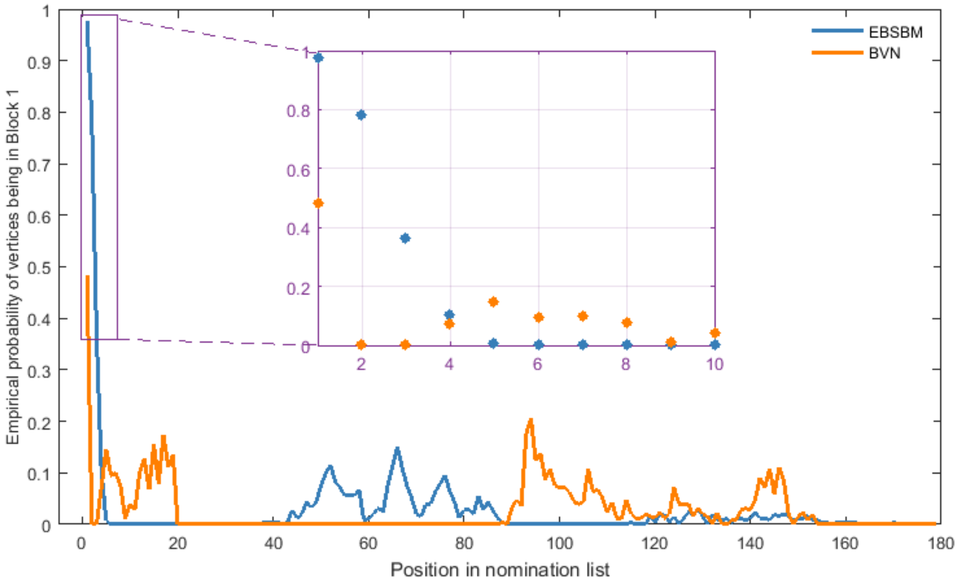


FIGURE 4.17: Empirical probabilities of vertices which are truly in block 1 for each position descending the nomination list for our Enron experiment.

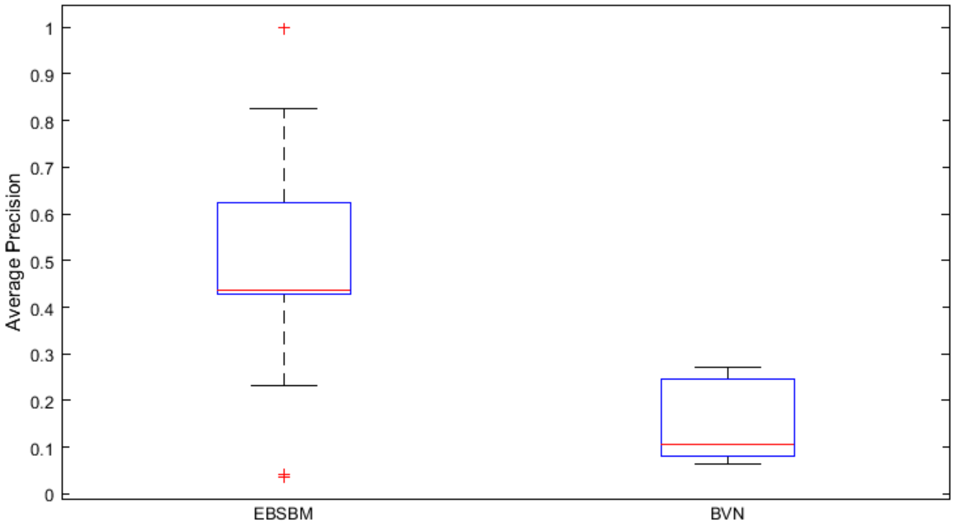


FIGURE 4.18: Boxplot of average precisions for our Enron experiment over 252 graph combinations.

probabilities in both models have the values of 0.8238 and 0.5914, correspondingly. Furthermore, the MRR of our method is 0.9767 with a 95% bootstrap CI of $(0.9533, 0.9922)$, which is substantially higher than 0.5734 with $(0.5226, 0.6244)$ as its 95% CI of BVN. Another useful insight is provided by the side-by-side boxplots of average precisions for both models as displayed in Figure 4.18. We observe from the boxplots that EBSBM does indeed give statistically significant performance advancement. A paired sample analysis produces a sign test p-value less than 10^{-34} for EBSBM against BVN. These models have an MAP of 0.4768 along with the associated 95 % CI of $(0.4533, 0.4998)$, and 0.1623 $(0.1519, 0.1729)$ for EBSBM and BVN, respectively; both of which are much better than an MAP of chance $(5/179 = 0.03)$. Again, EBSBM for this experiment has improved the nomination performance compared to BVN on this dataset.

4.6 SUMMARY

In this chapter the empirical Bayes model for vertex nomination, an extension to the novel empirical Bayes model introduced in Chapter 3 for block membership estimation of SBM, was proposed. This adaption is motivated by the original model's significant block assignment performance gains amongst the other contending models. Again, the proposed model in this chapter utilizes the Athreya et al. (2015) CLT for RDPGs (a generalization of SBM graphs) to obtain an empirical prior for the latent position parameter for an SBM. Similar to the previous chapter, inference with the model is conducted by implementing a Metropolis-within-Gibbs algorithm.

We conducted two simulation studies following the parameter values adopted from the earlier studies on vertex nomination for comparison purposes. While the first toy experiment in Section 4.4.1 followed a small-scale simulated graph originally considered by Lee et al. (2011), the second simulation study in Section 4.4.2 was based on the parameter values adopted in Coppersmith and Priebe (2012) on a larger graph. It is important to note that these earlier works are expected to give a better nomination performance than the new model proposed in this chapter. This is because both of these earlier models relied on a collective exploitation of information from attributed graphs with the presumption that in order to achieve better nomination performance exploiting both the graph structure as well as the edge attributes is useful. By contrast, the proposed model is operated on a partially observed non-attributed graph. Surprisingly, both the small-scale and large-scale simulation results demonstrate that our proposed model significantly yields a better nomination performance relative to chance and can be as effective as or better than its alternative competing approaches. Consistently good performance from the new model are also evident from the Enron graph experiment.

This chapter presented a new extension to the empirical Bayes model introduced in Chapter 3 for vertex nomination by exploiting information contained in the graph structure alone. The results we have thus far suggest that our modification of the empirical Bayes model is very effective and is a viable technique in addressing vertex nomination. Further developments of this method can potentially improve the nomination performance. Recent studies (Coppersmith and Priebe, 2012; Coppersmith, 2014; Suwan et al., 2015) show that information relevant to vertex nomination is present in both the graph structure and its edge attributes. Thus, a natural extension of our model utilizing both pieces of information derived from attributed graphs for this task is of theoretical and practical interest.

5 | EXTENDING VERTEX NOMINATION VIA EMPIRICAL BAYES ESTIMATION TO ATTRIBUTED GRAPHS

In most real-world networks, there often exists additional information about the vertices and edges beyond connectivity. This additional information can be incorporated into a network graph model as vertex and edge attributes, resulting in an attributed graph. The attributes can often be used to obtain improved solutions for problems involving network graphs. Examples include the social network graph, with vertices denoting individuals and their edges representing social interactions. We can have individuals' profile information including names, ages, or occupations embedded as vertex–attributes, and languages, communication topics, and the relationships' strength as the attributes on edges. Likewise, in the Wikipedia article graph, where manuscripts are vertices and hyperlinks between manuscripts are edges, an attribute on a vertex can be the topic assigned to the article by Wikipedia editors, while an attribute on an edge may be the address of the hyperlink. Another example in Bethard and Jurafsky (2010) uses an attributed graph for citation analysis. Here, vertices denote papers and edges denote citations between papers, and the edge attributes are a measure of text similarity between papers.

A myriad of studies have been devoted to analyzing vertex–attributed graphs, notably when information about a latent class membership is encoded as the vertex–attribute (e.g., stochastic blockmodels (Holland et al., 1983) and their extensions). However, graphs with edge attributes have a sparse literature apart from the use of scalar weights to indicate the closeness of the relationships between vertices (Aicher et al., 2014). The inclusion of attributes for both vertices and edges is of paramount interest as this information can potentially be powerful in a number of exploitation tasks including vertex nomination, thus being the motivation behind this chapter. Henceforth, in a slight abuse of terminology, we use the terms "attributed graph" and "edge-attributed graph" interchangeably.

As mentioned in Chapter 4, vertex nomination can be viewed as a subclass of recommender systems. At present, the methods for recommender systems on attributed graphs either place more emphasis on the structure of the graph (context) or on the attributes on the edges (content), rather than exploiting both of these together, even though the fusion of content and context has gained research momentum in recent years (see Section 5.1 for details).

In Chapter 3 we made use of recent theoretical results on spectral graph embedding to formulate an empirical Bayes approach for estimating block memberships in SBM graphs, which is essentially a community detection task. Since vertex nomination can be cast as a two-block SBM, we extended, in Chapter 4, our empirical Bayes method to perform vertex nomination on a partially observed non-attributed SBM graph. In this chapter, we further extend our vertex nomination model from Chapter 4 to utilize both context and content information from attributed SBM graphs. To facilitate this, we formulate a new likelihood model encapsulating the additional information from the edge attributes, together with the previous prior specifications for the model parameters from Chapter 3. This subsequently allows the construction of the ordered nomination list of vertices, prioritizing those that are most likely to be interesting.

The rest of the chapter is organized as follows. Section 5.1 provides a review and further details of various modeling approaches for attributed graphs currently featured in literature. Section 5.2 introduces notations and the vertex nomination problem setting for an attributed graph. Subsequently, descriptions of the extension of the empirical Bayes estimation and the posterior sampling scheme via a Metropolis–within–Gibbs sampler are given in Section 5.3. Section 5.4 presents and discusses a Monte Carlo simulation study which is based on the parameter settings originally employed in Coppersmith and Priebe (2012) as well as Section 4.4.2 of Chapter 4 for performance comparison. This is followed by Section 5.5 which is further divided into two parts, the first being the proposed extension of the EBSBM to the Enron corpus network. The second part is the simulation study which considers parameter settings closely emulating the Enron communication graph structure in order to understand the underlying cause for the lack of improvement in the nomination performance observed in the first part. We conclude with a summary of this chapter in Section 5.6.

5.1 Previous Work on Attributed Graphs

Most existing networks often contain both content and context-specific information which are useful for various inferential tasks. However, much of the earlier methods have leveraged on either one of the two. For instance, in the literature on recommender systems, Pavlov and Pennock (2002) developed a type of model-based recommendation method for a user's current navigation stream based on the context information. Their method also considered the sparsity and high-dimensionality of data by including a clustering of the articles depending on the user navigation arrangements. While Huang et al. (2005) formulated a link prediction approach by computing a set of linkage measures for each unlinked pair of user-items that aids in assessing the connection of two user-items for making recommendations. They showed that their approach yields promising results and suggested further exploration into this framework with the inclusion of other structural properties in graphs to improve the recommendation performance.

Besides a rapidly increasing interest of using information fusion from content and context by means of an attributed graph representation for various tasks, an attributed graph itself as a formalism to encode data is also becoming progressively prevalent across many disciplines. For instance, the structure of social networks, the Internet, and the physical structure of the brain are naturally well-suited to this representation. A myriad of studies have emerged from investigating the structure of the graph of such data and its corresponding inferential tasks including link structure prediction (Marchette and Priebe, 2008), citation retrieval (Bethard and Jurafsky, 2010), class membership estimations with applications to social networks (Eldardiry and Neville, 2012), protein-protein interactions, (Airoldi et al., 2006), connectomes (neuronal structure) (Vogelstein et al., 2013), and anomaly detection in social networks (Borges et al., 2011; Pao et al., 2011; Priebe et al., 2005).

At present, there is an overwhelming amount of literature on random graphs (Bollobás, 2001) and certain attributed graphs, for instance, vertex-attributed graphs, particularly when encoding a latent class as an attribute on a vertex (e.g. SBMs and their extension). However, there is sparse literature on graphs with both vertex-and edge-attributes. Bethard and Jurafsky (2010) studied a recommender system for scientific published articles by combining various content-based information that is essential to researchers such as author behavioral patterns, topic similarity between papers, the authors' previous paper citations, and centrality scores from the network of citations into a weighted linear scoring function for each probable citation and ranked the retrieved articles. Li and Zaïane (2004), in a similar manner, explored website recommendations by incorporating the websites' textual contents in conjunction with the connectivity information of web pages to construct user navigational models for recommendations, given a user's current status. Grothendieck et al. (2010) examined the benefit of using attributed graphs which encapsulate graph features and content of communications under the inference problem of detecting anomalous behavior in graphs. They formulated fusion tests from the likelihood ratio from attributed random graph models, and showed that the fusion of context- and content-based information can provide a more powerful inference than those relying on either context or content features alone. However, they also pointed out that this result is not always guaranteed as having a weak graph feature can reduce power. They further described the regions in parameter space where the fusion would provide the optimal performance by means of theoretical and numeric results. In a similar vein to this work, Gorin et al. (2010) discussed the joining of the content and context model on random attributed graphs and demonstrated how the joint model can provide a more effective statistical inference than either one used alone. Similarly this conclusion was reported by Priebe et al. (2010) who experimented on time series of Enron data based on random attributed graphs for the anomaly detection task. They illustrated that a statistic that includes both content and context information can give better inference when compared to those statistics which only include one of the two. This work is then further explored by Brinda et al. (2011) who performed simple hypothesis tests about the Erdös–Rènyi graph on a fixed number of vertices and random edges where each of these edges can only have exactly one attribute. Similarly, Tang et al. (2013a) explored the anomaly detection problem on a time-series of attributed graphs. This was accomplished by using the moving average based test statistics of some graph invariants. They derived the limiting distribution of these test statistics assuming that the number of vertices is sufficiently large, and in turn derived the estimation of the power of the tests.

In the context of SBM, Nowicki and Snijders (2001) proposed a posteriori stochastic blockmodeling for digraphs where they provided the frameworks for integrating edge attributes information into the model which they referred to as alphabets. This work was an extension of Snijders and Nowicki (1997) to the situation where interactions of vertex pairs can be directed with arbitrary possible values of the number of blocks. They modelled interactional structure conditional on the vertex-attributes by using a generalization of the SBM. A Bayesian paradigm based on Gibbs sampling was employed to approximate the posterior distribution over the model parameters and posterior predictive distribution. As the vertices' class assignments are dependent on a vector of vertex-attribute parameter which is not observed, this model can be conceived as a mixture model. Thus, they consequently encountered the problem of non-identifiability of parameters. This is tackled by restricting inference to the posterior distributions of those invariant functions of parameters and the latent attributes with respect to class relabellings.

A graph with weighted edges can also be viewed as an attributed graph where weight on the edges often captures extra information about vertices' interactions such as their interaction frequency, character, volume, or strength of the relationship. Aicher et al. (2014) introduced the weighted stochastic blockmodel for a community detection task. They integrated information about both the existence and weight of the edges into the model in order to learn the graph's structure. They derived a variational Bayes algorithm to estimate the model parameters. Through simulation studies and real-world applications they showed how their proposed model allows the discovery of block membership structures in a broader spectrum of edge-weighted networks without having to discard weight information; a process of which was required in the classical SBM.

Recently, Coppersmith and Priebe (2012) introduced the notion of vertex nomination, a primary exploitation task in this thesis (see Chapter 4 for details), using attributed graphs. Their model is based on two main assumptions which are believed to exist in networks that contain groups of interest. Specifically, the assumptions are

1. Communications among pairs of vertices within the interesting group (known and unknown) are more intense than other pairs.

2. The communication topics amongst interesting vertices are distinct from the rest of the graph.

The first assumption indicates that the information for vertex nomination can be determined from the context, while the content information can be deduced from the second assumption. Following the notation of the previous chapter in Section 4.2, they considered an attributed graph, each of which is colored red (interesting) or green (uninteresting), but only m' vertices are observed to be red. The color of the other vertices is unobserved. Each edge is also colored red or green and this is observed for all edges. For each vertex i, they defined its context statistic as the number of observed red vertices connected to i, and its content statistic as the number of red edges incident to i. Assuming that these statistics are independent between vertices and that red edges are more likely between red vertices, Coppersmith and Priebe proposed a likelihood model with a simple linear fusion of these statistics to rank the hidden vertices for nomination. The model is then applied to synthetic experiments as well as on the Enron email dataset where the performance of the model was measured by a number of tests, namely the probability of correct nomination, mean reciprocal rank, and mean average precision, as stated in Section 4.3.3. They showed that even using a simple fusion of content and context information can provide as well as or better nomination performance than when each is used alone.

In a similar spirit as Coppersmith and Priebe (2012), a Bayesian vertex nomination using content and context statistics on an attributed graph as defined by Coppersmith and Priebe was later proposed by Lee et al. (2011). They modified Coppersmith and Priebe (2012)'s likelihood together with prior distributions chosen for the unknown parameters and unobserved vertex colors. The procedure considered using MCMC via a Metropolis-within-Gibbs algorithm to efficiently sample from the posterior distribution of vertex colors given statistics related to the observed graph applied to simulation studies and the Enron dataset. This method demonstrated a clear advantage over the existing methods and has many potential applications in social network analysis and graph inference (see Section 5.4 for a comparison with our model).

Moreover, Marchette et al. (2011) extended an RDPG model (a special case of the latent position model) as discussed in Section 2.3.5.1 to incorporate the edge attribution function for vertex nomination. A latent position associated to each vertex was estimated by adopting the iterative approach stated in Algorithm 1 of Scheinerman and Tucker (2010). The model assumed that the presence of an edge and edge attributes are fundamentally connected, where the dimension of the latent positions is set to be equal to the number of edge-attributes (i.e. assigning one dimension per edge attribute). In this model, the probability of an edge between any vertex pair is a dot product of the corresponding latent positions, and the probability of an edge having attribute k is proportional to the value of weight given to those dimensions in the associated latent positions. The nomination performance was evaluated based on two measures, the first, a probability of the top-ranked candidate vertex, is truly interesting and the second is the normalized sum of reciprocal ranks. Results from a simulation study as well as from the Enron data experiment suggested that the attributed RDPG is another viable method for the vertex nomination problem and it is worth pursuing further.

Within the multimedia retrieval paradigm, the efficacy of unifying content and context information in order to discover the underlying latent semantic space for imperative subsequent analysis was also reported by Qi et al. (2012b). They used attributed graphs to encode multimedia information networks with vertices representing multimedia objects and edges as the content links (referred to as the sound and/or visual similarities among objects). In addition, they further enriched the graph representation by including context objects (e.g. user-generated tags) and their corresponding attributes (i.e. textures, colors). The method heavily relied on the notion of a latent position model of Hoff et al. (2002), where each vertex is embedded into a latent vector in a low dimensional space which intrinsically encodes the information in context and content links. Given these latent vectors, multimedia objects can then be adequately indexed, classified, and retrieved by employing conventional multimedia retrieval approaches (i.e. the support vector machine (Shawe-Taylor and Cristianini, 2004) or clustering).

5.2 Attributed Graph Representation for Vertex Nomination

The vertex nomination model introduced in Chapter 4 is formulated as a two–block SBM without edge-attribute. Identical to Chapter 4, a random graph, G, is assumed to be undirected with no self–loops, multi-edges or hyper–edges. Recall that the block of interesting/red vertices is $\mathcal{M}$ with $|\mathcal{M}| = m \ll n$, and hence the block of uninteresting/green vertices is $V \setminus \mathcal{M}$. As in Chapter 4, we assume that the block membership of vertices is unobserved except for a small number, m', of interesting vertices in $\mathcal{M}'$. Thus, $|\mathcal{M}'| = m'$, $\mathcal{M}' \subset \mathcal{M}$, $V \setminus \mathcal{M}'$ contains all those vertices whose block membership is unobserved, $\mathcal{M} \setminus \mathcal{M}'$ contains those interesting vertices whose block membership is unobserved. We also continue to assume that the constraints, $1 < m' \leq m \ll n$, hold.

In a slight deviation from the previous chapter, in this chapter we allow the edges in E to have a categorical attribute taking value $l \in \mathcal{L}$. Each $e \in E$ is represented by a triple $(i, j, l) \in V \times V \times \mathcal{L}$. In our setting, $(i, j, l) \in E$ is a communication between a vertex pair, i and j, and having attribute l belonging to the set $\mathcal{L}$.

More specifically, for the vertex nomination task, an edge attribute can only take values, "interesting/red" or "uninteresting/green", implying $|\mathcal{L}| = 2$. We suppose that the attribution of edges are observed for all edges. This may be encapsulated by an $n \times n$ edge–attributed adjacency matrix, $S \in \{0, 1, 2\}^{n \times n}$, as follows: S_{ij} between vertex i and j is 1 if a red edge is present, $S_{ij} = 2$ if a green edge is present, and $S_{ij} = 0$ if no edge exists.

Figure 5.1 depicts our model setting and the vertex classifications. Bright red circles represent observed interesting vertices, light green ($V \setminus \mathcal{M}$) and red ($\mathcal{M} \setminus \mathcal{M}'$) circles represent those vertices with unobserved block–labels. The dashed boarders signify vertices' true block memberships.

5.3 Model

In this section, we describe how our empirical Bayes model can be extended to exploit edge attributes for vertex nomination. Marchette et al. (2011) extended the RDPG model of Young and Scheinerman (2007) to incorporate information about edge attributes. Their model is constructed by allowing the existence of an edge and its attributes to be interrelated. Specifically,

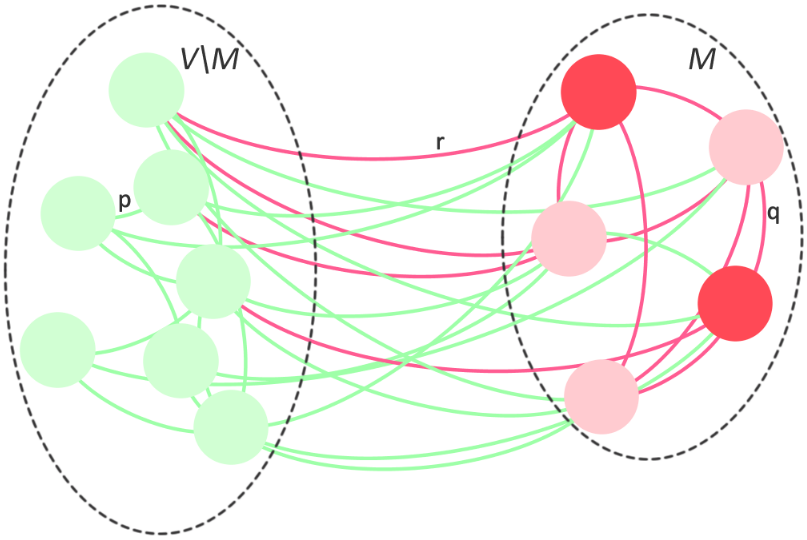


FIGURE 5.1: Illustrative example of the attributed graph model setting for vertex nomination. Here, $m' = 2$ vertices are the known interesting/red vertices, $m - m' = 3$ are the unobserved interesting/red vertices, and $n - m = 7$ are the unobserved uninteresting/-green vertices. Edges denote communication between connected vertices. In addition, edge attributes denote content of communication which is either red or green (1 or 2, accordingly). The edge attribute between two red vertices is governed by the probability vector $q = (q_0, q_1, q_2)$, while between two green is governed by $p = (p_0, p_1, p_2)$, and $r = (r_0, r_1, r_2)$ is for the edge attribute between a green and a red one.

the edges in their model represent communications between vertices, with communication topics encoded by the attributes, and the latent positions in their RDPG model encode the level of interest of each vertex in each topic. To facilitate this, Marchette et al. (2011) assigned one latent position dimension to each edge attribute.

However, for our vertex nomination task, each edge has only one attribute (i.e. either interesting/red or uninteresting/green) indicating the content of the communication. As such, we deviate slightly from Marchette et al. (2011) by introducing a pair of latent positions for each vertex. Specifically, in the context of an RDPG model stated in Section 2.3.5.1, vertex i is assigned latent positions, X_i and Y_i, corresponding to a red edge and a green edge respectively. Thus, for vertices i and j, the probability of a red edge is $\langle X_i, X_j \rangle$, the probability of a green edge is $\langle Y_i, Y_j \rangle$, and the probability of no edge is $1 - \langle X_i, X_j \rangle - \langle Y_i, Y_j \rangle$.

For the two-block SBM with a red block and a green block, there will be two distinct pairs of latent positions (ν_1, ω_1) and (ν_2, ω_2), with the first pair for the red block and the second for the green one.

Along the lines of Coppersmith and Priebe (2012) and Suwan et al. (2015), we define the probability vectors, $q = (q_0, q_1, q_2)$, $p = (p_0, p_1, p_2)$, and $r = (r_0, r_1, r_2)$, which govern, respectively, the edge probabilities within the red block, within the green block, and between the two blocks.

For instance, within the red block, q_1 is the probability of a red edge, q_2 is the probability of a green edge, and thus q_0 is the probability of no edge; similarly for p and r (see Figure 5.1).

Thus, given the block memberships, $\tau_1, \ldots, \tau_n$, and the latent positions, $\nu = (\nu_1, \nu_2)$ and $\omega = (\omega_1, \omega_2)$, we can explicitly express these probabilities as

$$\begin{aligned} \text{i}\quad & \mathbb{P}(S_{ij} = 1 \mid \tau_i = \tau_j = 1, \nu) = \langle \nu_1, \nu_1 \rangle = q_1, \\ & \mathbb{P}(S_{ij} = 2 \mid \tau_i = \tau_j = 1, \omega) = \langle \omega_1, \omega_1 \rangle = q_2, \end{aligned}$$

$$\begin{aligned} \text{ii}\quad & \mathbb{P}(S_{ij} = 1 \mid \tau_i = \tau_j = 2, \nu) = \langle \nu_2, \nu_2 \rangle = p_1, \\ & \mathbb{P}(S_{ij} = 2 \mid \tau_i = \tau_j = 2, \omega) = \langle \omega_2, \omega_2 \rangle = p_2, \end{aligned}$$

$$\begin{aligned} \text{iii}\quad & \mathbb{P}(S_{ij} = 1 \mid \tau_i \neq \tau_j, \nu, \omega) = \langle \nu_1, \nu_2 \rangle = r_1, \\ & \mathbb{P}(S_{ij} = 2 \mid \tau_i \neq \tau_j, \nu, \omega) = \langle \omega_1, \omega_2 \rangle = r_2. \end{aligned}$$

Clearly, when ignoring the edge attributes, the probability of an edge between two red vertices would be $q_1 + q_2$, between two green vertices is $p_1 + p_2$, and between a red and a green vertex is $r_1 + r_2$. Our definition of the edge probability vectors is different from Coppersmith and Priebe (2012) and Suwan et al. (2015) where the edge probability between two green vertices is assumed equal to the edge probability between a red vertex and a green one, i.e. $p = r$.

The two key assumptions established in both Coppersmith and Priebe (2012) and Suwan et al. (2015) concerning vertex nomination are

1. connections between vertices in the interesting block, $\mathcal{M}$, both observed ($\mathcal{M}'$) and unobserved ($\mathcal{M} \setminus \mathcal{M}'$), are at a different frequency than other vertex pairs in $V \setminus \mathcal{M}$,
2. an edge between two vertices in $\mathcal{M}$ has a different probability of being interesting than for a pair of vertices in $V \setminus \mathcal{M}$ or a pair of vertices between $\mathcal{M}$ and $V \setminus \mathcal{M}$.

The first assumption suggests that information pertaining to block assignment of vertices can be obtained from the structure of the graph, while the second suggests that this information can be obtained from the edge attributes. For our model, we consider a special case of these assumptions in the form of the following constraints: $r_1 + r_2 \leq p_1 + p_2 < q_1 + q_2$, and $r_1 \leq p_1 < q_1$. The first constraint says that more connections can be expected within the interesting block, and the second says that more interesting connections can be expected within the interesting block. However, for ease of exposition, in this chapter we shall assume that $p_2 = q_2 = r_2$, which satisfy the first constraint, $r_1 + r_2 \leq p_1 + p_2 < q_1 + q_2$, trivially when the second constraint, $r_1 \leq p_1 < q_1$ is satisfied.

Conditional on the block membership vector, $\tau = (\tau_1, \ldots, \tau_n)$, and the latent vectors in ν and ω, the new likelihood function modified from Eqn 4.1 to incorporate the information about edge attributes is

$$\begin{aligned} f(S \mid \tau, \nu, \omega) = & \prod_{i=1}^{m'} \prod_{j>i}^{m'} \{1 - (\langle \nu_1, \nu_1 \rangle + \langle \omega_1, \omega_1 \rangle)\}^{\mathbb{I}_{\{0\}}(S_{ij})} \cdot \langle \nu_1, \nu_1 \rangle^{\mathbb{I}_{\{1\}}(S_{ij})} \cdot \langle \omega_1, \omega_1 \rangle^{\mathbb{I}_{\{2\}}(S_{ij})} \\ & \prod_{i=1}^{m'} \prod_{j>m'}^{n} \left\{1 - \left(\langle \nu_1, \nu_{\tau_j} \rangle + \langle \omega_1, \omega_{\tau_j} \rangle\right)\right\}^{\mathbb{I}_{\{0\}}(S_{ij})} \cdot \langle \nu_1, \nu_{\tau_j} \rangle^{\mathbb{I}_{\{1\}}(S_{ij})} \cdot \langle \omega_1, \omega_{\tau_j} \rangle^{\mathbb{I}_{\{2\}}(S_{ij})} \\ & \prod_{i=m'+1}^{n} \prod_{j>i}^{n} \left\{1 - \left(\langle \nu_{\tau_i}, \nu_{\tau_j} \rangle + \langle \omega_{\tau_i}, \omega_{\tau_j} \rangle\right)\right\}^{\mathbb{I}_{\{0\}}(S_{ij})} \cdot \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{\mathbb{I}_{\{1\}}(S_{ij})} \cdot \langle \omega_{\tau_i}, \omega_{\tau_j} \rangle^{\mathbb{I}_{\{2\}}(S_{ij})}. \end{aligned} \tag{5.1}$$

The posterior distribution can be defined in the usual way as proportional to the likelihood in Eqn (5.1) multiplied by the prior distributions of the unknown quantities, τ, ν, and ω. The prior distribution for τ is the same multinomial distribution as before (see Section 3.2.1). In order to formulate our empirical Bayes prior distributions for the parameters ν and ω, we adopt the ASGE technique discussed in Chapters 3 and 4 with a slight modification to estimate the latent positions, $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$.

Let $G^{(a)} = (V, E^{(a)})$ be the subgraph of G such that $E^{(a)}$ is the set of edges with edge attribute a. Let the binary adjacency matrix associated with $G^{(a)}$ be $S^{(a)} \in \{0, 1\}^{n \times n}$. Recall that in our case, a can either take a value of 1/red, or 2/green. It is reported in Marchette et al. (2011) that by decomposing an edge–attributed adjacency matrix in this manner and operating on $S^{(1)}$ and $S^{(2)}$ independently, some information about the original attributed graph is lost. However, this loss can be ignored on when a graph is large. Whilst this approach to attributed graph decomposition is convenient, the resulting subgraphs can be sparse. To circumvent this, we use the spectral clustering with perturbations approach (SCP) proposed by Amini et al. (2013), which was shown to work well on sparse graphs. The method proceeds by computing the empirical average vertex degree, $\lambda^{(a)}$, for each adjacency matrix, $S^{(a)}$, then adding the perturbation, $\lambda^{(a)}/(4n)$, to the elements in $S^{(a)}$.

We then employ the ASGE technique discussed in Section 2.4, to embed $S^{(1)}$ and obtain the estimates of the latent positions, $\widehat{X}_1, \ldots, \widehat{X}_n$. The subsequent step is to fit a two-component GMM on these estimates, $\widehat{X}_i$, via `MCLUST` as discussed in Section 2.5. Similar to Chapter 4, the mixture component means, $\widehat{\nu} = (\widehat{\nu}_1, \widehat{\nu}_2)$, and covariance matrices, $\widehat{\Sigma}_\nu = \left(\widehat{\Sigma}_{\nu_1}, \widehat{\Sigma}_{\nu_2}\right)$, given by the GMM fit, $\widehat{X}_i$, are used for the empirical Bayes prior for ν. A similar procedure is applied separately to $S^{(2)}$ to yield the estimates for the latent positions, $\widehat{Y}_1, \ldots, \widehat{Y}_n$. Recall that for ease of exposition, we impose the condition, $p_2 = q_2 = r_2$, which we satisfy by further assuming that $\omega_1 = \omega_2$. Thus, we fit a one-component GMM on these estimates, $\widehat{Y}_i$, and obtain the mean, $\widehat{\omega}_1$, and covariance matrix, $\widehat{\Sigma}_{\omega_1}$, which are used for the empirical Bayes prior for ω_1.

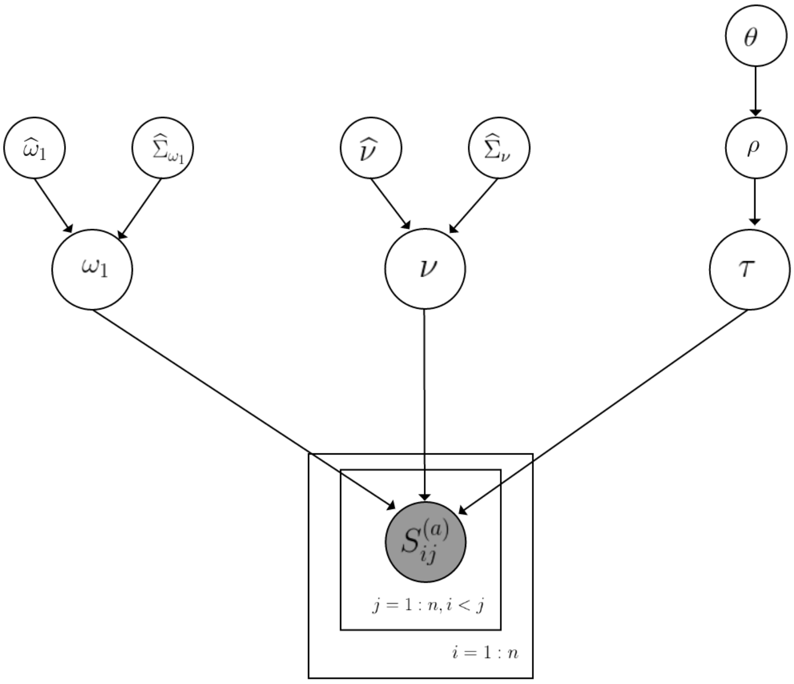


Figure 5.2: Hierarchical structure of the empirical Bayes model given in Section 5.3 represented as a DAG.

Since the parameter, ν, has to comply with assumption (2) for vertex nomination, that is, $r_1 \leq p_1 < q_1$, ν is constrained to be in the set :

$$\mathcal{S}_\nu = \{\nu : 0 \leq \langle \nu_1, \nu_2 \rangle \leq \langle \nu_2, \nu_2 \rangle < \langle \nu_1, \nu_1 \rangle \leq 1\}. \tag{5.2}$$

To ensure that the dot product involving ω_1 is a probability, ω_1 is constrained to be in the set,

$$\mathcal{S}_{\omega_1} = \{\omega_1 : \langle \omega_1, \omega_1 \rangle \in [0, 1]\}, \tag{5.3}$$

In addition, since the edge existence probabilities between vertices in two-block SBM are $q_1 + q_2$, $p_1 + p_2$, and $r_1 + r_2$, it is necessary to also ensure that the sum of the two dot products involving ν and ω_1 are probabilities. We thus have

$$\mathcal{S}_{\omega_1|\nu} = \{\omega_1 : \langle \nu_1, \nu_1 \rangle + \langle \omega_1, \omega_1 \rangle, \langle \nu_2, \nu_2 \rangle + \langle \omega_1, \omega_1 \rangle, \langle \nu_1, \nu_2 \rangle + \langle \omega_1, \omega_1 \rangle \in [0, 1]\}. \tag{5.4}$$

Put together, the prior distributions for the unknown quantities, τ, ν, and ω_1 are

$$\begin{aligned}
\tau \mid \rho &\sim \text{Multinomial}(\rho), \\
\rho &\sim \text{Dirichlet}(\theta), \\
\nu \mid \widehat{\nu}, \widehat{\Sigma}_\nu &\sim \mathbb{I}_{\mathcal{S}_\nu}(\nu) \prod_{k=1}^{2} \mathcal{N}_d \left(\nu_k \mid \widehat{\nu}_k, \widehat{\Sigma}_{\nu_k} \right), \\
\omega_1 \mid \nu, \widehat{\omega}, \widehat{\Sigma}_\omega &\sim \mathbb{I}_{\mathcal{S}_{\omega_1}}(\omega_1) \mathbb{I}_{\mathcal{S}_{\omega_1 \mid \nu}}(\omega_1) \mathcal{N}_d \left(\omega_1 \mid \widehat{\omega}_1, \widehat{\Sigma}_{\omega_1} \right).
\end{aligned}$$

In a similar fashion as Chapters 3 and 4, we marginalize the posterior distribution over ρ since a conjugate Dirichlet prior is placed on ρ, resulting in

$$\begin{aligned}
f(\tau, \nu, \omega_1 \mid S) &\propto f(S \mid \tau, \nu, \omega_1) \cdot f(\tau \mid \theta) \cdot f(\omega_1 \mid \nu, \widehat{\omega}_1, \widehat{\Sigma}_{\omega_1}) \cdot f(\nu \mid \widehat{\nu}, \widehat{\Sigma}_\nu) \\
&\propto f(S \mid \tau, \nu, \omega_1) \cdot \left[\prod_{k=1}^{2} \Gamma(\theta_k + T_k) \right] \cdot f(\omega_1 \mid \nu, \widehat{\omega}_1, \widehat{\Sigma}_{\omega_1}) \cdot f(\nu \mid \widehat{\nu}, \widehat{\Sigma}_\nu),
\end{aligned}$$

where $T = (T_1, T_2)$ again is the block membership counts as defined in Section 3.2.1. Note that the resulting posterior distribution for this model is slightly different to that of Eqn (4.4), due to the modification of the likelihood and the prior specifications for ν and ω_1. Forthwith, we will refer to this model as the attributed EBSBM or AEBSBM for short. Figure 5.2 represents the proposed model's hierarchical structure.

5.3.1 The Posterior Sampling Scheme

Posterior sampling for the extended model is achieved via MCMC, again using a Metropolis–within–Gibbs algorithm as in Chapter 4. To update the block membership vector, τ, a standard Gibbs update is performed employing its full–conditional distribution,

$$\begin{aligned}
f(\tau_i | \tau_{-i}, S, \nu, \omega, \theta) \propto & \prod_{j=m'+1, j \neq i}^{n} \left\{ 1 - \left(\langle \nu_1, \nu_{\tau_j} \rangle + \langle \omega_1, \omega_{\tau_j} \rangle \right) \right\}^{\mathbb{I}_{\{0\}}(S_{ij})} \cdot \langle \nu_1, \nu_{\tau_j} \rangle^{\mathbb{I}_{\{1\}}(S_{ij})} \cdot \langle \omega_1, \omega_{\tau_j} \rangle^{\mathbb{I}_{\{2\}}(S_{ij})} \\
& \prod_{j \neq i} \left\{ 1 - \left(\langle \nu_{\tau_i}, \nu_{\tau_j} \rangle + \langle \omega_{\tau_i}, \omega_{\tau_j} \rangle \right) \right\}^{\mathbb{I}_{\{0\}}(S_{ij})} \cdot \langle \nu_{\tau_i}, \nu_{\tau_j} \rangle^{\mathbb{I}_{\{1\}}(S_{ij})} \cdot \langle \omega_{\tau_i}, \omega_{\tau_j} \rangle^{\mathbb{I}_{\{2\}}(S_{ij})} \\
& \left[\prod_{k=1}^{K} \Gamma(\theta_k + T_k) \right],
\end{aligned} \tag{5.5}$$

where $\tau_{-i} = \tau \setminus \tau_i$ denotes all the block memberships of vertices excluding vertex i. This consists of visiting each τ_i, for $i = 1, \dots, n$, and executing Algorithm 6 by first initializing $\tau^{(0)} = \widehat{\tau}$, which is the GMM clustering solution of $\widehat{X}$. Notably, the conditional posterior distribution for τ_i given $\tau_{-i}, S, \nu, \omega, \theta$ is Bernoulli(ρ_i^*) where ρ_i^* can be expressed as

$$\rho_{i,k}^* = \frac{f(\tau_i = k \mid \tau_{-i}, S, \nu, \omega, \theta)}{\sum_{k'=1}^{K} f(\tau_i = k' \mid \tau_{-i}, S, \nu, \omega, \theta)}, k = 1, 2. \tag{5.6}$$

Algorithm 6 Gibbs sampling of the block assignment vector τ

```
At iteration $t$;
for $i = 1, \dots, n$ do
    Compute $\rho_i^*(\tau_1^{(t)}, \dots, \tau_{i-1}^{(t)}, \tau_{i+1}^{(t-1)}, \tau_n^{(t-1)})$ as in Eqn (5.6)
    Set $\tau_i^{(t)} = k$ with probability $\rho_{i,k}^*$
end for
```

M-H sampling is carried out for each Gibbs step to update ν and ω_1. For ν, akin to Chapter 4, the step consists of initializing ν by its empirical Bayes prior, $\nu^{(0)} | \widehat{\nu}, \widehat{\Sigma}_v \sim f(\nu \mid \widehat{\nu}, \widehat{\Sigma}_\nu)$, followed by proposing an update of $\widetilde{\nu}$, where again the prior for ν will be used as the proposal distribution. Once proposed, they are accepted or rejected using the usual M-H accept/reject probability.

The M-H sampler for ω_1 will be identical to ν in the sense that the empirical Bayes prior for ω_1, $f(\omega_1 \mid \nu, \widehat{\omega}_1, \widehat{\Sigma}_{\omega_1})$, will be employed in both the initial, $\omega_1^{(0)}$, and proposal state, $\widetilde{\omega}_1$. Once proposed, we set $\widetilde{\omega}_2 = \widetilde{\omega}_1$. The details are outlined in Algorithm 7.

Algorithm 7 Metropolis–Hasting update of the latent positions, ν and ω.

```
At iteration $t$;
for $m = 1$ to $10$ do
    Propose $\widetilde{\nu} \sim \mathbb{I}_{\mathbb{S}_\nu}(\nu) \prod_{k=1}^{2} \mathcal{N}_d\left(\nu_k \mid \widehat{\nu}_k, \widehat{\Sigma}_{\nu_k}\right)$
    Propose $\widetilde{\omega}_1 \sim \mathbb{I}_{\mathbb{S}_{\omega_1}}(\widetilde{\omega}_1)\mathbb{I}_{\mathbb{S}_{\omega_1|\nu}}(\widetilde{\omega}_1)\mathcal{N}_d(\widetilde{\omega}_1 \mid \widetilde{\nu}, \widehat{\omega}_1, \widehat{\Sigma}_{\omega_1})$
    Set $\widetilde{\omega} = (\widetilde{\omega}_1, \widetilde{\omega}_1)$
    Accept $\nu^{(t)} = \widetilde{\nu}$ and $\omega^{(t)} = \widetilde{\omega}$ with probability, $\min(1, \alpha)$, where
        $$\alpha = \frac{f(S \mid \tau^{(t)}, \widetilde{\nu}, \widetilde{\omega})}{f(S \mid \tau^{(t)}, \nu^{(t-1)}, \omega^{(t-1)})};$$
    Otherwise set $\nu^{(t)} = \nu^{(t-1)}$ and $\omega^{(t)} = \omega^{(t-1)}$
end for
```

Note that the additional edge information in the graph has resulted in the model having extra parameters to estimate (when compared to the vertex nomination model of Chapter 4), thus it can be computationally more expensive. See below for further comparisons.

5.4 Simulation Study

In this section, we simulate random attributed graphs adopting the values that Coppersmith and Priebe (2012) used. The results are then compared and contrasted with competing approaches including the model in Chapter 4 (abbreviated as EBSBM) and the BVN model of Suwan et al. (2015).

Recall that for ease of exposition, we impose the constraints, $p_2 = q_2 = r_2$, to ensure that the probability of presenting a green edge is identical throughout the graph, and $q_1 > p_1 \geq r_1$ to ensure that the probability of a red edge presenting is higher for edges appearing within $\mathcal{M}$ than the rest of the graph. These constraints imply that $q_0 < p_0 \leq r_0$, indicating the probability of the presence of an edge is higher among vertices in $\mathcal{M}$ than other pairs outside it. For a random experiment, we thus fix p, q, and r to be

$$q = (0.4, 0.4, 0.2), p = (0.6, 0.2, 0.2), r = (0.6, 0.2, 0.2).$$

In the context of the two-block SBM, edge probabilities are determined by the entries of matrix B. We have $q_1 + q_2$ governing the presence of an edge between two red vertices (i.e. within block 1), while $p_1 + p_2$ governs the presence of an edge between two green vertices (i.e. within block 2), and $r_1 + r_2$ governs the probability of an edge existing between a red and a green vertex. Thus, with the values chosen above, $p_1 + p_2 = r_1 + r_2$. Therefore, the edge probability matrix, B, and the block proportion vector, ρ, are

$$B = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.4 \end{pmatrix} \qquad \text{and} \qquad \rho = \left(\frac{m}{n}, \frac{n-m}{n} \right). \tag{5.7}$$

Note that the resulting edge probability matrix is identical to that in Section 4.4.2.

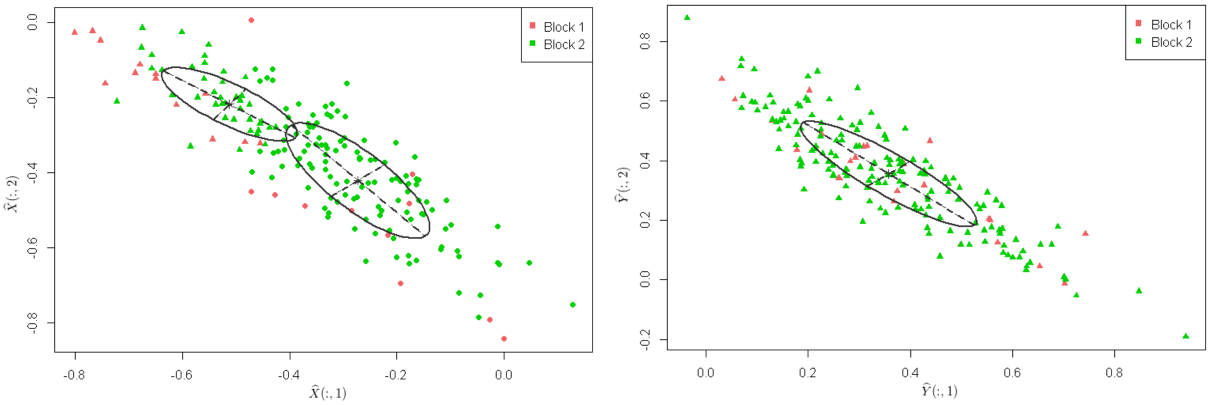
Analogous to Section 4.4.2, the probability mass function of the latent positions, (X_i, Y_i), is a mixture of point masses:

$$f(X_i, Y_i \mid \rho, \nu, \omega) = \sum_{k=1}^{2} \rho_k \delta_{\nu_k, \omega_k}(X_i, Y_i),$$

where $(\nu_1, \omega_1) \approx ((-0.6155, 0.1453), (0.4472, 0))$ and $(\nu_2, \omega_2) \approx ((-0.3804 - 0.2351), (0.4472, 0))$.

As in Section 4.4.2, we set $n = 184$ and consider varying the total number of red vertices, m, and the number of known red vertices, m', which again satisfy the constraints, $1 < m' \leq m \ll n$.

Particularly, we have three set of ratios: $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$ with $m \in \{8, 12, \ldots, 36\}$ in increments of 4. For each combination of m and m', we generate 1000 Monte Carlo graph replicates on n vertices.



(A) $\widehat{X}_i$ (B) $\widehat{Y}_i$

FIGURE 5.3: An illustration of clustering solutions from the GMM procedure of $\widehat{X}_i$ and $\widehat{Y}_i$ for a graph with $n = 184$, $m = 24$ and $m' = 6$. The colors represent the true block memberships, whereas symbols represent the clustering solutions given by the GMM. The ellipses represent the 95% confidence region for the two cluster latent vectors of the estimated GMM. The axes correspond to the $d = 2$ dimensions of the data.

Figure 5.3 gives scatter plots of the estimated latent positions, $\widehat{X}_i$ and $\widehat{Y}_i$, for one of the Monte Carlo replicates with $m = 24$ and $m' = 6$. The symbols denote the clustering outputs from the GMM procedure, and the colors denote the true classes. The ellipses again represent the 95% confidence region for the two cluster latent vectors of the estimated GMM.

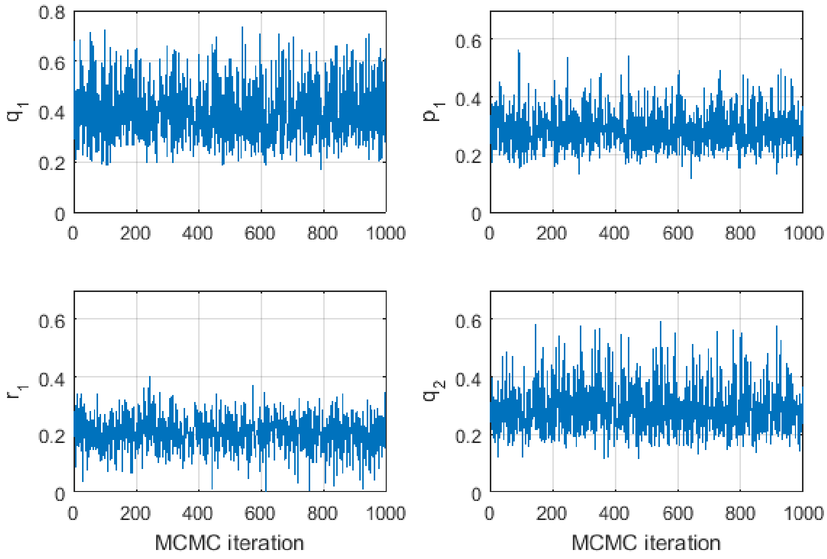


FIGURE 5.4: Trace plots of the edge attribute probabilities, q_1, p_1, r_1, and $q_2 = p_2 = r_2$, as defined in Section 5.3. Recall that the true parameter values are 0.4, 0.2, 0.2, and 0.2, correspondingly.

For each graph replicate, the sampling routine described in Section 5.3.1 is run for two parallel Markov chains for a large number of iterations from the posterior distribution until convergence. Again, we use the Gelman-Rubin diagnostic test explained in Section 3.4.1.2 to indicate convergence of the MCMC chains. Posterior inference for τ is then based on a collection of the last 500 iterations from both chains giving a total of 1000 MCMC sample points. Trace plots

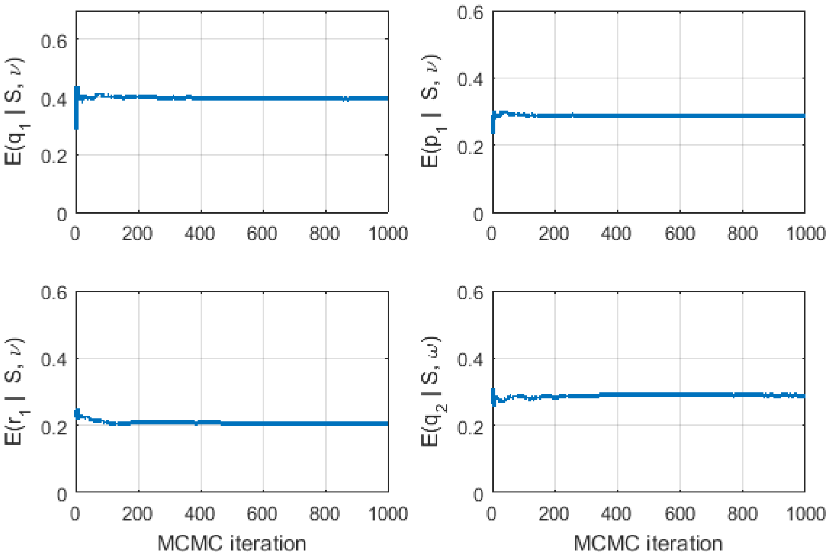


FIGURE 5.5: Trace plots of cumulative average estimates of the marginal posterior means of the edge attribute probabilities, q_1, p_1, r_1, and $q_2 = p_2 = r_2$.

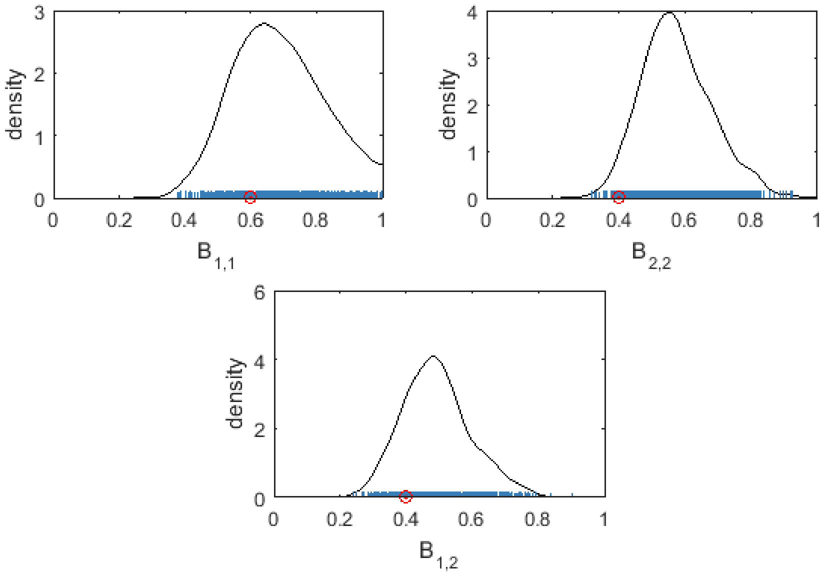


FIGURE 5.6: Posterior densities estimates for the entries in matrix B. Red points on the horizontal axis denote the true parameter values.

of the edge attribute probability vectors, p, q, and r, and cumulative average estimates of their marginal posterior means computed using Eqn (4.12) for $m = 24, m' = 6$, are displayed as an example in Figure 5.4 and Figure 5.5, respectively. This is followed by Figure 5.6 which provides the marginal posterior densities of the edge probabilities within and between blocks 1 and 2 for the same example. It can be seen from the marginal posterior density plots that the true parameter values of B (red points) are located approximately around the concentration of posterior densities. Similar findings can be seen from Figure 5.7, which shows an illustration of kernel density estimates for the posterior means for p, q, and r across 1000 graph replicates where $m = 24, m' = 6$.

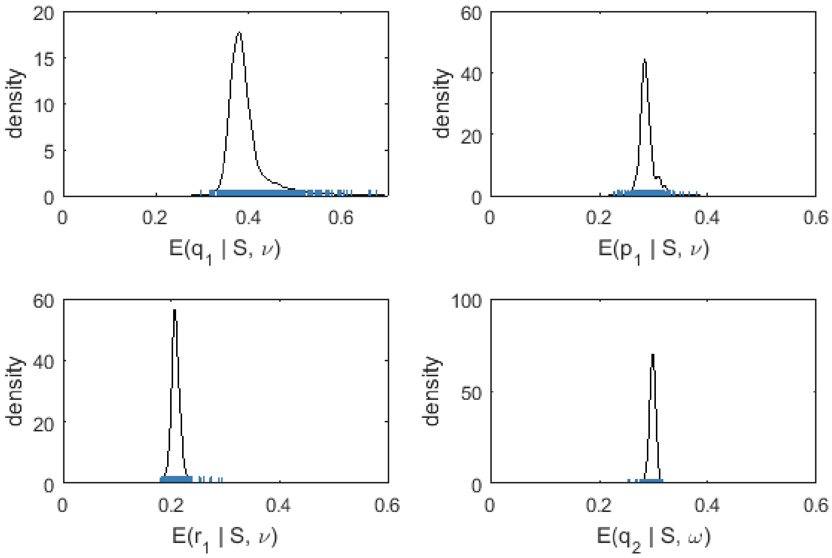
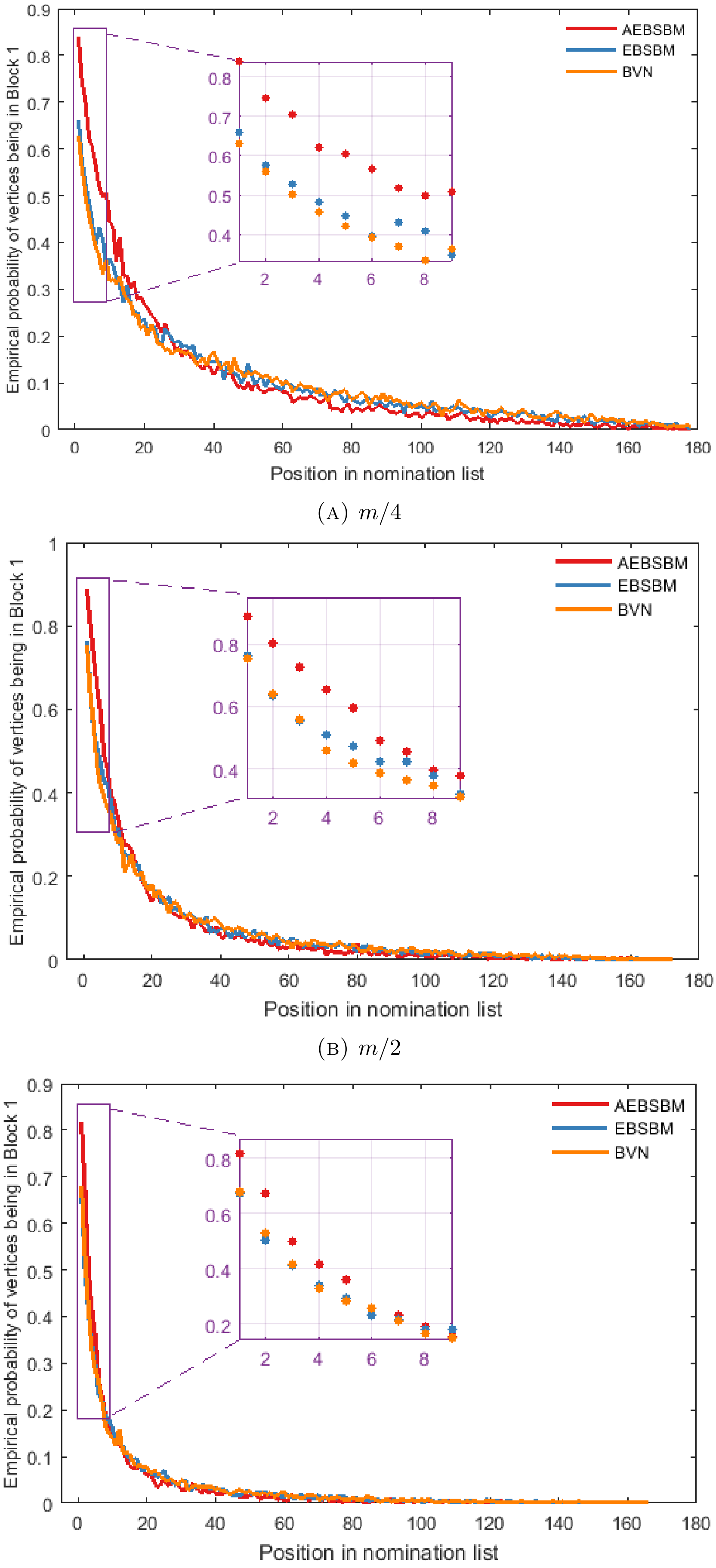


FIGURE 5.7: Kernel density estimates for posterior means of q_1, p_1, r_1, and $q_2 = p_2 = r_2$, in the case of $m = 24, m' = 6$ obtained over 1000 graph replicates. As a reminder, the true parameter values are 0.4, 0.2, 0.2, and 0.2, respectively.

For brevity, in Figure 5.8, the empirical probabilities of vertices being in the red block are plotted against the nomination list position, for the case of $m = 24$ and $m = 6$ in all three sets (from top to bottom: $\frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$) as a demonstration. These can be computed according to the description outlined in Section 4.4.1. For comparison purposes, the results from EBSBM as well as BVN are also included.

Since the model formulated in this chapter is based on exploiting an attributed graph, which is a step-up in complexity from the previous chapter, it is expected that a strong difference would appear between our two vertex nomination models. Unsurprisingly, Figure 5.8 shows that AEBSBM generally demonstrates exceedingly superior nomination performance among the contenders near the beginning of the nomination list. This is evident from the AEBSBM's higher values of empirical probabilities of vertices being truly red. Specifically, in the case of $m/4$ (the first subplot), AEBSBM gives the best performance in the first top three positions of the nomination list, whereas EBSBM and BVN seem to be very competitive throughout with EBSBM displaying a better performance in nominating a single red vertex. However, as the position in



(A) $m/4$

(B) $m/2$

(C) $3m/4$

FIGURE 5.8: Comparison of empirical probabilities of vertices being a member of the block of interest against their position in the nomination list obtained from 1000 graphs, where $n = 184, m = 24$ for all three sets for the AEBSBM (blue),EBSBM (red) and BVN (green). Columns, from top to bottom, represent $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$.

the nomination list descends, all three models give an approximately similar performance. While in the case of $m/2$, having half of the red vertices in the block known, AEBSBM continues to win against the other contenders in the first two positions of the list, but after the third position downwards all models' performance recede in a similar fashion. Lastly, similar results are evident when 75% of vertices in the red block are known (i.e. $3m/4$), particularly, AEBSBM again yields the highest empirical probability of the first positioned vertex being a true red.

Identical to Section 4.4.2, we also employ the evaluation criteria explained in Section 4.3.3 to assess the nomination performance of AEBSBM and compare it with EBSBM and BVN. Firstly, the right-hand side of Figure 5.9 shows the comparison of the competing models using MRR values. Recall that this measure is for assessing the efficacy of the method in correctly identifying one more red vertex. Generally, MRR values in this figure agree with the earlier visual inspections of the nomination performance via the illustration plots of empirical probabilities of vertices being in the red block in Figure 5.8. That is, AEBSBM yields results vastly superior to EBSBM and BVN; e.g., for $m = 24$, $m' = 6$, the MRR for AEBSBM is approximately 0.9042 with a 95% CI of $(0.8900, 0.9176)$ compared to the MRR for EBSBM of approximately 0.8010 with $(0.7830, 0.8190)$ as its 95% CI. Based on the paired sample analysis, for instance, in the same example, the sign-test p-value is less than 10^{-23}. Despite the low MRR values when m is small, e.g. when $m = 8$, in all three sets, MRRs are no greater than 0.34 for AEBSBM and 0.26 for EBSBM; a continuous improvement in the performance is evident in all three simulation sets as m increases (MRRs approach 1). This indicates that these models are better at positioning the truly red vertex at the top of a ranked nomination list. AEBSBM produces the most outstanding performance using this measure out of the three models. A strong difference in performance can be seen in the middle plot of Figure 5.9 on the right-hand side where the ratio is $m' = m/2$. However, when m gets larger than 36 the gaps between AEBSBM and the other two competing methods reduce immensely.

Moreover, when assessing the models based on the MAP from the same figure on the left-hand side, similar remarks can still be made. AEBSBM yields a statistically significantly better nomination performance than EBSBM and BVN for all three experiment sets. This is confirmed by the paired sample analysis again using the sign-test which yields p-values less than 10^{-8} for AEBSBM vs EBSBM and 10^{-5} for AEBSBM vs BVN; except for the case when $m = 8, m' = 2$, and the p-value is 0.0469. As a summary, Table 5.1 reports the MAP and MRR values together with the respective 95% bootstrap confidence intervals for AEBSBM and EBSBM.

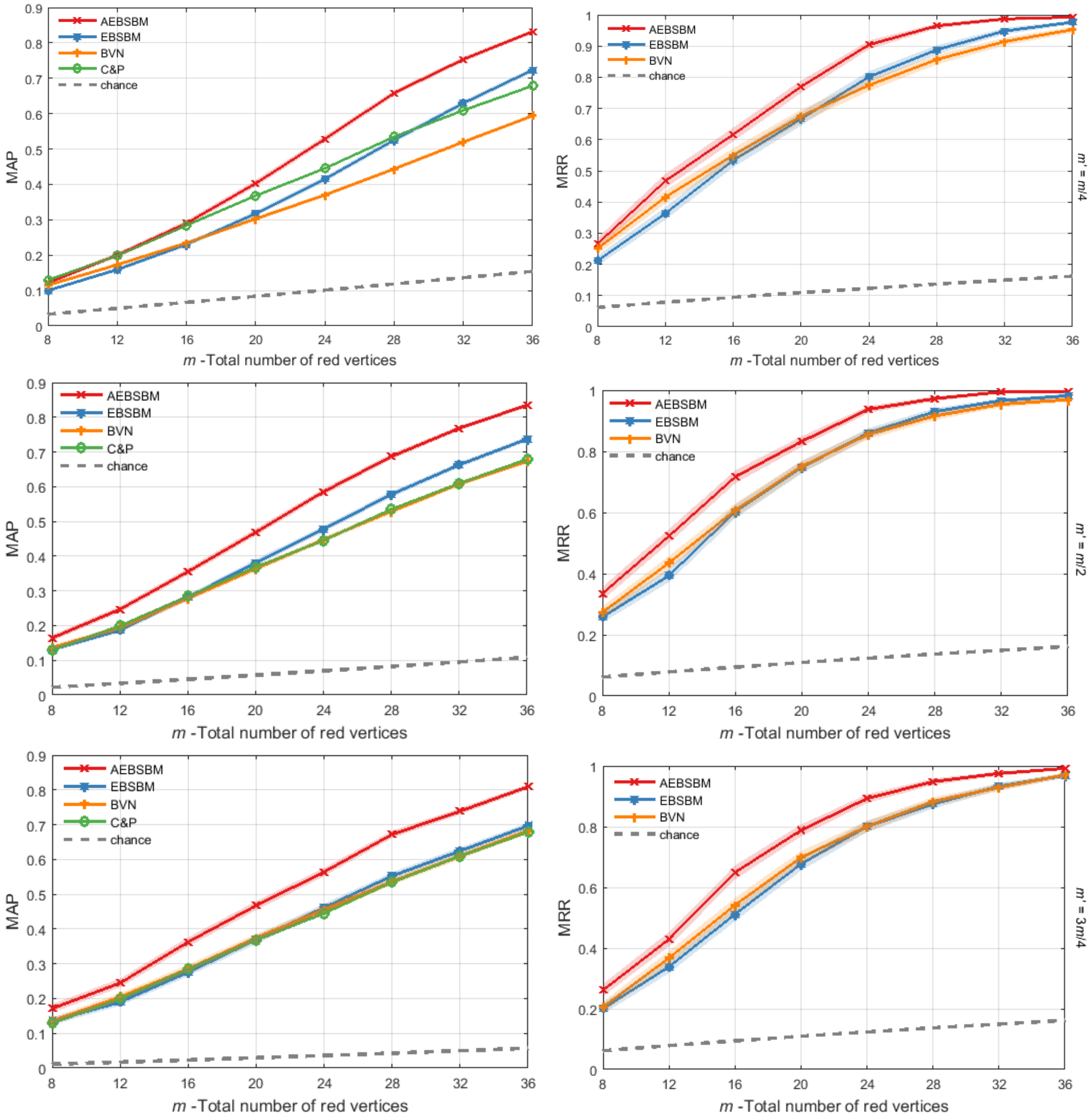


FIGURE 5.9: The nomination performance according to mean average precision (MAP) and minimum reciprocal rank (MRR) (*left* and *right*, respectively) across three sets of ratios. Rows, from top to bottom, indicate $m' = \frac{m}{4}, \frac{m}{2}, \frac{3m}{4}$. The results are obtained from 1000 graph realizations. The x-axis represents increasing values of m and the y-axis represents MAP and MRR values, correspondingly. Shaded areas represent 95% bootstrap standard errors.

Table 5.1: MAP and MRR with the associated 95% confidence intervals given in the parenthesis for the simulation experiment in this Chapter.

Model		$m' = m/4$							
		8	12	16	20	24	28	32	36
EBSBM	MAP	0.0998	0.1593	0.2303	0.3174	0.4152	0.5254	0.6296	0.7233
	95% CI	[0.0950, 0.1050]	[0.1530, 0.1650]	[0.2240, 0.2370]	[0.3100, 0.3250]	[0.4070, 0.4230]	[0.5170, 0.5340]	[0.6220, 0.6380]	[0.7160, 0.7300]
	MRR	0.2120	0.3630	0.5340	0.6660	0.8010	0.8880	0.9480	0.9760
	95% CI	[0.1960, 0.2290]	[0.3430, 0.3840]	[0.5110, 0.5560]	[0.6450, 0.6880]	[0.7830, 0.8190]	[0.8740, 0.9030]	[0.9380, 0.9580]	[0.9690, 0.9830]
AEBSBM	MAP	0.1214	0.2004	0.2901	0.4028	0.5283	0.6584	0.7534	0.8320
	95% CI	[0.1158, 0.1271]	[0.1932, 0.2079]	[0.2817, 0.2987]	[0.3930, 0.4126]	[0.5189, 0.5376]	[0.6500, 0.6668]	[0.7458, 0.7608]	[0.8261, 0.8376]
	MRR	0.2661	0.4673	0.6165	0.7700	0.9042	0.9649	0.9871	0.9927
	95% CI	[0.2481, 0.2846]	[0.4445, 0.4902]	[0.5944, 0.6387]	[0.7502, 0.7894]	[0.8900, 0.9176]	[0.9561, 0.9728]	[0.9814, 0.9918]	[0.9879, 0.9962]
		$m' = m/2$							
EBSBM	MAP	0.1310	0.1870	0.2810	0.3810	0.4780	0.5780	0.6630	0.7370
	95% CI	[0.1240, 0.1380]	[0.1790, 0.1950]	[0.2720, 0.2900]	[0.3720, 0.3910]	[0.4680, 0.4880]	[0.5690, 0.5870]	[0.6550, 0.6710]	[0.7300, 0.7440]
	MRR	0.2580	0.3940	0.6030	0.7480	0.8600	0.9300	0.9660	0.9820
	95% CI	[0.2400, 0.2770]	[0.3720, 0.4150]	[0.5810, 0.6260]	[0.7280, 0.7680]	[0.8440, 0.8760]	[0.9170, 0.9410]	[0.9570, 0.9740]	[0.9760, 0.9870]
AEBSBM	MAP	0.1643	0.2467	0.3550	0.4684	0.5852	0.6869	0.7685	0.8349
	95% CI	[0.1555, 0.1733]	[0.2375, 0.2562]	[0.3449, 0.3652]	[0.4581, 0.4790]	[0.5754, 0.5950]	[0.6782, 0.6955]	[0.7614, 0.7753]	[0.8290, 0.8406]
	MRR	0.3348	0.5225	0.7168	0.8327	0.9373	0.9723	0.9941	0.9960
	95% CI	[0.3134, 0.3565]	[0.4997, 0.5457]	[0.6954, 0.7380]	[0.8153, 0.8495]	[0.9257, 0.9481]	[0.9643, 0.9794]	[0.9900, 0.9971]	[0.9920, 0.9980]
		$m' = 3m/4$							
EBSBM	MAP	0.1350	0.1900	0.2760	0.3690	0.4620	0.5520	0.6240	0.6970
	95% CI	[0.1240, 0.1460]	[0.1790, 0.2010]	[0.2640, 0.2870]	[0.3570, 0.3810]	[0.4500, 0.4730]	0.5420, 0.5630	[0.6140, 0.6350]	0.6880, 0.7060
	MRR	0.2010	0.3390	0.5120	0.6770	0.8000	0.8750	0.9330	0.9690
	95% CI	[0.1830, 0.2190]	[0.3170, 0.3600]	[0.4890, 0.5350]	[0.6550, 0.6990]	[0.7820, 0.8190]	[0.8590, 0.8900]	[0.9210, 0.9440]	[0.9600, 0.9760]
AEBSBM	MAP	0.1711	0.2445	0.3627	0.4671	0.5641	0.6719	0.7387	0.8090
	95% CI	[0.1589, 0.1837]	[0.2320, 0.2576]	[0.3495, 0.3760]	[0.4541, 0.4800]	[0.5522, 0.5761]	[0.6614, 0.6823]	[0.7292, 0.7480]	[0.8012, 0.8165]
	MRR	0.2613	0.4292	0.6494	0.7881	0.8927	0.9483	0.9753	0.9920
	95% CI	[0.2414, 0.2819]	[0.4060, 0.4532]	[0.6260, 0.6723]	[0.7685, 0.8072]	[0.8778, 0.9065]	[0.9372, 0.9584]	[0.9677, 0.9821]	[0.9870, 0.9950]

5.5 Enron Data

This section is divided into two parts. Firstly, Section 5.5.1 looks at the application considered both in Suwan et al. (2015) and Chapter 4 to illustrate the extended vertex nomination model. While Section 5.5.2 considers another simulation study with the parameter values highlighting the network structure of the Enron dataset.

5.5.1 Application to Enron Graphs

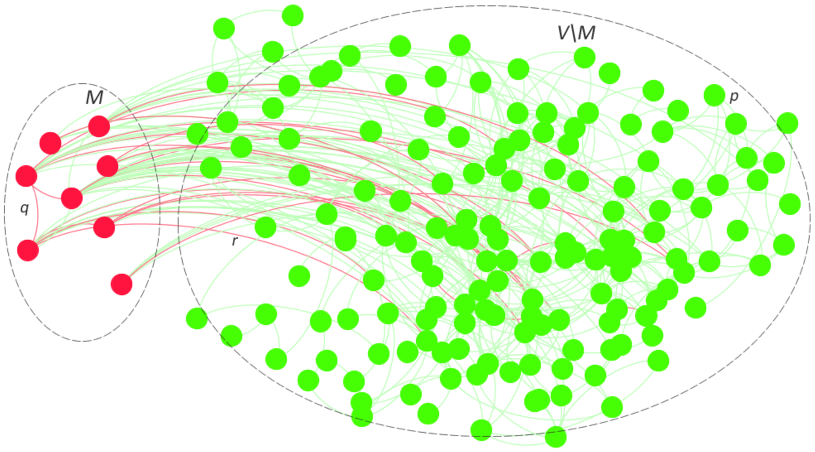
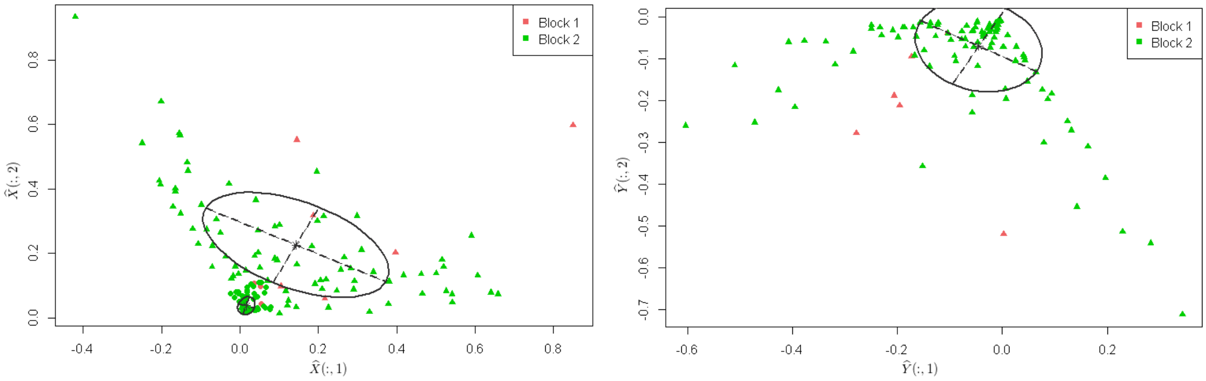


Figure 5.10: Our Enron graph after excluding isolated vertices, with $n = 184$ vertices: $m = 10$ for Class 1 = red = fraudsters, $n - m = 174$ for Class 2 = green = non-fraudsters.

AEBSBM is applied to week 38 and 58 of Priebe et al. (2005)'s Enron graphs. Berry et al. (2001) categorized the contents of the emails into 32 topics, then later Coppersmith and Priebe (2012) mapped these topics into a binary edge attribute, {red, green}, which correspondingly represent the content of messages as dubious and indubious. We use the Enron graphs derived by Priebe et al. (2005) in conjunction with the binary edge attributes from Coppersmith and Priebe (2012), to demonstrate the efficacy of our method using the AEBSBM. Recall that each graph contains 184 vertices, 10 of which have been identified as fraudsters.

Similar to Section 4.5, we partition vertices into $K = 2$ classes, *fraudsters*/red and *non-fraudsters*/green (i.e. $\{\mathcal{M}, V \setminus \mathcal{M}\}$), assuming that only 5 out of 10 fraudsters are known. We then generate 252 possible combination of graphs (i.e. 5 observed red vertices taken from 10 fraudsters). Figure 5.10 displays one rendering of the Enron graph in week 38. Here, edges denote communication between connected vertices, and edge attributes signify the content of the communication; both of which are observed.

For each edge-attributed graph a pre-processing routine is carried out in an identical manner

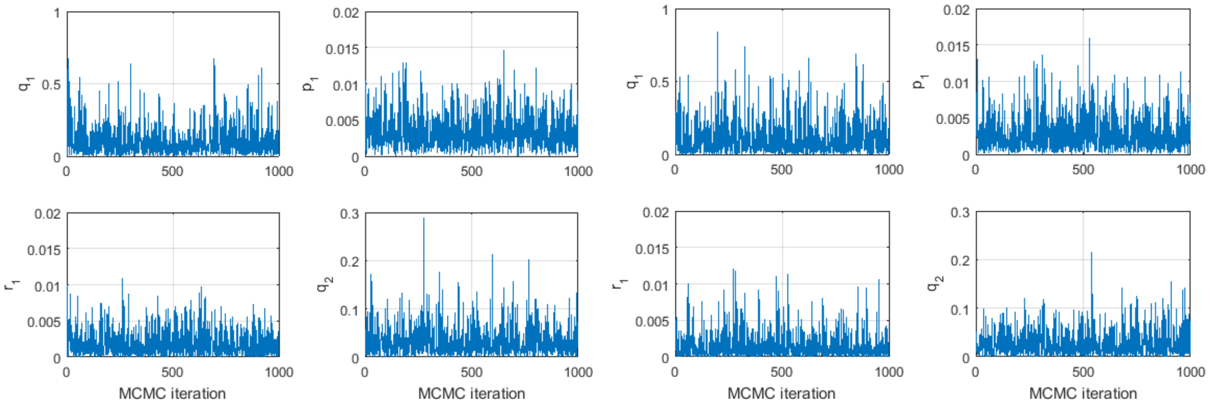


(A) $\widehat{X}_i$

(B) $\widehat{Y}_i$

FIGURE 5.11: An illustration of clustering solutions from the GMM procedure of $\widehat{X}_i$ and $\widehat{Y}_i$ for one of the resulting graphs in week 38. The colors represent the true block memberships, whereas symbols represent the clustering solutions given by the GMM. The ellipses denote the 95% confidence region for the two cluster location vectors of the estimated GMM. The axes correspond to the $d = 2$ dimensions of the data.

to that of Section 5.4 to obtain the empirical priors for the pair of latent position parameters. Figure 5.11 provides the adjacency spectral embeddings $\widehat{X}_i$ and $\widehat{Y}_i$ for one of the graph combinations with colors denoting their true classes, and symbols denoting the clustering solutions given by the GMM. Subsequently, the posterior sampling routines as outlined in Section 5.3.1

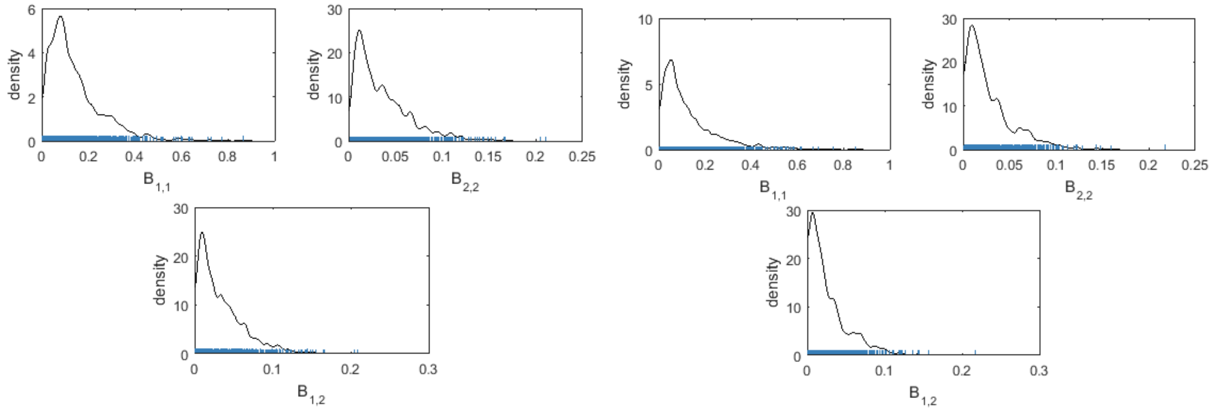


(A) Week 38

(B) Week 58

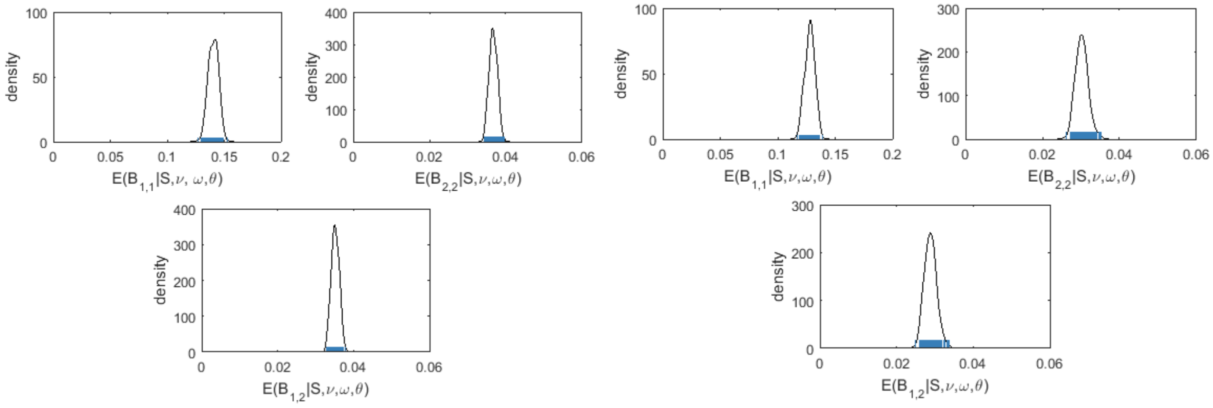
FIGURE 5.12: Trace plots of parameter vectors $p = (p_0, p_1, p_2)$, $q = (q_0, q_1, q_2)$, and $r = (r_0, r_1, r_2)$, for one combination of the Enron graph in week 38 (*left*) and week 58 (*right*). Note that $q_2 = p_2 = r_2$, thus only the trace plot of q_2 is shown for brevity.

are executed for posterior inference. Figure 5.12 exhibits trace plots of the edge attribute probability vectors, p, q, and r, while Figure 5.13 displays the marginal posterior densities for the components in the edge probability matrix, B, and Figure 5.14 gives the distributions of the marginal posterior means of the components in B from the 252 combinations. Plots of week 38 are displayed on the left side and week 58 are on the right. The visual inspection of trace plots and the Gelman-Rubin assessment of convergence as discussed in Section 3.4.1.2, indicate that no evidence of lack of convergence was found. As a reminder, we do not directly estimate the parameters, p, q, r and B, but rather estimate the latent positions associated to a red and a green edge (i.e. ν, and ω), respectively, which are then be parametrized using the dot product kernel as previously explained in Section 5.3.



(A) Week 38 (B) Week 58

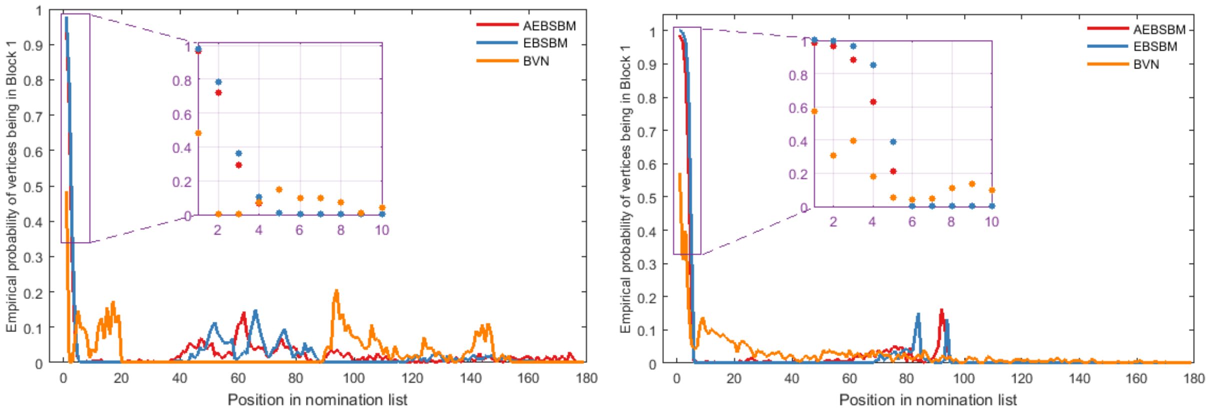
FIGURE 5.13: Marginal posterior densities for the edge probability matrix B for one combination of the Enron graph in week 38 (*left*) and week 58 (*right*).



(A) Week 38 (B) Week 58

FIGURE 5.14: Kernel density estimates for posterior means of the edge probability matrix, B, obtained over 252 graph combinations in week 38 (*left*) and week 58 (*right*).

Similar to Section 4.5, the empirical probabilities of membership of vertices in the red block for the respective 179 positions in the nomination list are given in Figure 5.15. For the experiment on the Enron graph in week 38, we see that the empirical probabilities of vertices being truly red in the first position of both AEBSBM and EBSBM are close to 1, indicating a very competitive nomination performance, while BVN is much smaller with a probability of less than 0.5. When observing the second position in the nomination list the probability for the AEBSBM appears to drop slightly below 0.7. While in the case of EBSBM, although the probability also decreases from the first position it is still slightly higher than AEBSBM with a value of 0.78. However,



(A) Week 38

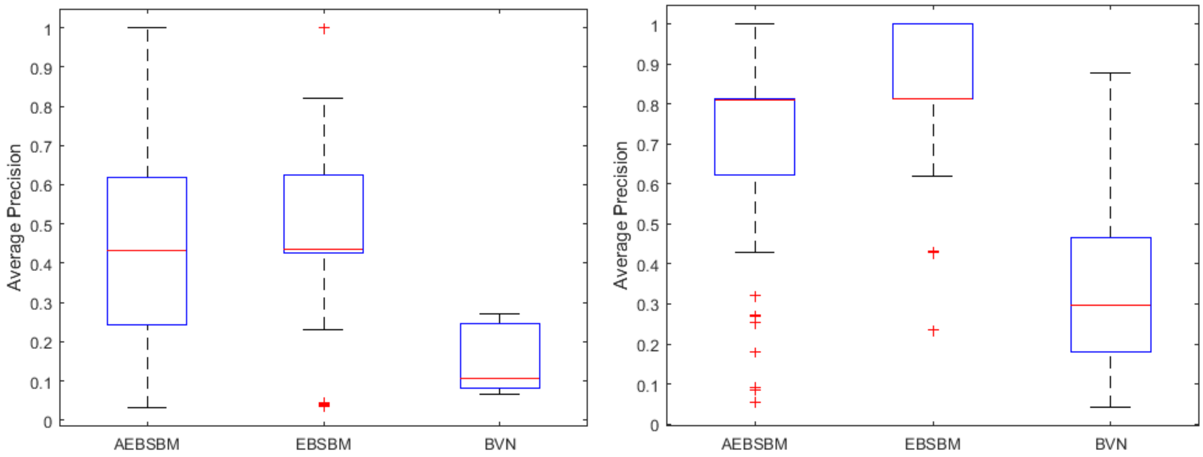
(B) Week 58

Figure 5.15: Empirical probabilities of vertices classified as a member of the group that committed fraud against their position in the nomination list obtained across 179 graphs for BVN (green), EBSBM (orange), and AEBSBM (blue). Right-hand side figures represent week 38 and left-hand side figures represents week 58.

after the third position onwards, it appears that these probabilities promptly descend to nearly 0 for all models. This visual inspection of the nomination performance is supported by looking at the MRR. In the same week, the MRR of AEBSBM is 0.9666 with a 95% bootstrap CI of $(0.9425, 0.9857)$, where the value lies within a 95% CI of EBSBM with an MRR of 0.9767 and $(0.9534, 0.9922)$ as its 95% CI. This indicates that these two models are comparable to each other when the task focuses on identifying one more fraudster. As previously shown in Section 4.5 that EBSBM is significantly superior to BVN, undoubtedly, AEBSBM also shows superiority. These similar findings also appeared in week 58 where the MRR scores of both models appear significantly competitive with each other with the score of 1 and 0.9861, accordingly, where in this case EBSBM obtains a perfect MRR score. The sign test p-values for the paired-sample replicates are 0.3438 and 0.1250 for week 38 and week 58, respectively.

It is also evident in both weeks that AEBSBM gives undeniably superior nomination performance than BVN, but surprisingly somewhat less so against EBSBM. The MAP of AEBSBM is 0.4414 and $(0.4189, 0.4641)$ as its 95% CI, while the MAP of EBSBM is 0.4768 ($(0.4533, 0.4998)$) for week 38. Our paired sample analysis gives the sign-test p-value of less than 10^{-9} for AEBSBM against EBSBM Nonetheless, all methods are better than the MAP of chance (i.e. 0.03). Similarly, in the case of week 58, although generally MAPs of all models are higher than week 38 presumably due to a more densely connected graph, EBSBM yields a significantly better nomination performance than AEBSBM (the sign-test p-value is less than 10^{-15}). Specifically, the MAP scores are 0.7507 with $(0.7266, 0.7739)$ as its 95% CI for AEBSBM, 0.8466 together with a 95% CI of $(0.8286, 0.8639)$ for EBSBM and lastly 0.3260 with $(0.3047, 0.3476)$ for BVN.

Assessment of the models can also be performed visually using boxplots summarizing simulated AP values given in Figure 5.16. It appears from the boxplots that the week 38 experiment contains no outliers for AEBSBM and a few outliers for EBSBM. AEBSBM is seen to be roughly symmetrical, whereas EBSBM and BVN boxplots show a positive skew. Further, there is substantially more variation in AEBSBM with AP ranges approximately from 0.03 to 1. Contrarily, EBSBM ranges are approximately from 0.23 to 1 (an exclusion of outliers), and BVN from 0.07 to 0.27. The spread of the middle 50% of the ordered AP values is twice as large for AEBSBM as for EBSBM, again suggesting that the AP value is more variable for AEBSBM. In the case of week 58, all models overall have higher AP scores than in week 38. AEBSBM still exhibits high variability, however the shape in this case is left skewed with a few outliers. Similar to week 38, the distribution of the AP for EBSBM is somewhat skewed to the right and appears to be concentrated in the upper quantile of AEBSBM. As for BVN, the bulk distribution in both experiments lies significantly below the lower quantile of both AEBSBM and EBSBM, although the boxplot of week 58 appears to be more spread out than week 38. Also see Table 5.2 for a summary of the MAP and MRR values of the three competing methods along with their associated 95% bootstrap CI.



(A) Week 38

(B) Week 58

FIGURE 5.16: Box plots of average precisions for our Enron experiment over 252 graphs for both week 38 (*left*) and week 58 (*right*).

Unfortunately, when the AEBSBM is applied to the Enron data, the nomination performance does not show a significant improvement from the EBSBM as expected, despite the additional inclusion of edge attribute information to the AEBSBM. The premise that information relevant to vertex nomination can be found in the edge attributes, we translate it into the constraint set, $q_1 > p_1 \geq r_1$, as defined in Section 5.3. This precisely implies that the edges between the red vertices have higher attributes than other pairs of vertices within the green block and between the red and

Table 5.2: MAP and MRR with the associated 95% confidence intervals given in the parenthesis for the Enron data .

Model	Week 38		
	MRR	MAP	$\pi(k)$
AEBSBM	0.9666 $[0.9425, 0.9857]$	0.4414 $[0.4189, 0.4641]$	0.4103 $[0.3865, 0.4333]$
EBSBM	0.9767 $[0.9534, 0.9922]$	0.4768 $[0.4533, 0.4998]$	0.4468 $[0.4214, 0.4698]$
BVN	0.5734 $[0.5218, 0.6245]$	0.1623 $[0.1519, 0.1729]$	0.1405 $[0.1278, 0.1508]$
	Week 58		
AEBSBM	0.9861 $[0.9696, 0.9970]$	0.7507 $[0.7266, 0.7739]$	0.7349 $[0.7095, 0.7595]$
EBSBM	1 $[1, 1]$	0.8466 $[0.8286, 0.8639]$	0.8373 $[0.8190, 0.8556]$
BVN	0.7040 $[0.6599, 0.7470]$	0.3260 $[0.3047, 0.3476]$	0.3024 $[0.2825, 0.3206]$

green blocks. However, the empirical edge-attribute probability vectors for the Enron graph in week 38 are $\bar{q} = (0.8667, 0.1333, 0)$, $\bar{p} = (0.9738, 0.0191, 0.0071)$, and $\bar{r} = (0.9546, 0.0333, 0.0121)$; clearly in this case, $r_1 > p_1$. We believe that this violation of the AEBSBM's assumption somewhat degrade the nomination performance (when compared to the EBSBM). A similar violation of the model assumption is also evident in week 58, where $\bar{q} = (0.7556, 0.1333, 0.1111)$, $\bar{p} = (0.9727, 0.0236, 0.0037)$, and $\bar{r} = (0.9425, 0.0506, 0.0069)$, respectively.

This rationale is supported in the simulation study which will be discussed in the next section. Random graphs with the same number of vertices, total number of red vertices, and number of observed red vertices (i.e. n, m, and m', correspondingly) as the Enron data, are generated using the parameters which satisfy all the model's constraints, except for $r_1 > p_1$.

5.5.2 Enron Simulation Study

The simulation study in this section assesses the performance for the AEBSBM when applied to an edge-attributed two-block SBM graph with parameter values illuminating the Enron graph structure. Specifically, for a random experiment, we set

$$q = (0.4, 0.4, 0.2), p = (0.7, 0.2, 0.1), r = (0.6, 0.3, 0.1).$$

Consequently, the components in the edge probability matrix, B, are

$$B = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.3 \end{pmatrix} \quad \text{and} \quad \rho = \left(\frac{m}{n}, \frac{n-m}{n} \right). \tag{5.8}$$

Coppersmith (2014) asserted that information with regard to the classification of vertices can be attained from the edge attributes, if edges amongst red vertices possess different attributes from the rest of the graph. Thus, as part of the AEBSBM formulation, we considered a special case of this assumption in the form of the constraints, $q_1 > p_1 \geq r_1$. This section aims to investigate how a violation of the aforementioned assumption impacts the resulting posterior inference for the vertex nomination task. More precisely, for the values chosen above, $r_1 > p_1$.

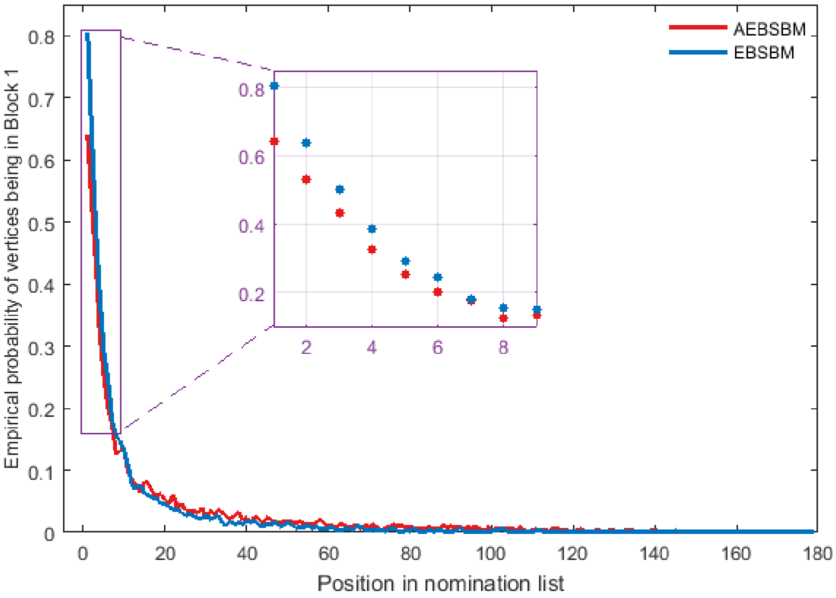


FIGURE 5.17: Plot of empirical probabilities of vertices which are truly classified as a member in the red block obtained from 1000 realizations with $n = 184, m = 10, m' = 5$ for AEBSBM (blue) and EBSBM (orange).

For $n = 184$, $m = 10$, and $m' = 5$, we independently generate 1000 random attributed graphs. For each graph replicate, an identical posterior sampling routine, as explained in Section 5.4, is executed for AEBSBM and the performance of the model is assessed via the evaluation measures discussed in Section 4.3.3. Particularly, AEBSBM yields an MAP of 0.4949 with a 95% CI of $(0.4755, 0.5076)$, while the MAP of EBSBM is 0.5950 with $(0.5818, 0.6082)$ as its 96% CI, indicating the superiority of EBSBM over AEBSBM (the sign test p-value for the paired Monte Carlo replicates is less than 10^{-16}). Figure 5.17 illustrates the nomination performance via the empirical posterior probability of classifying vertices as red for each vertex descending the ranked nomination list. As expected, the empirical posterior probability of a vertex being truly red decreases as the nomination list position increases for both models. More importantly, it is evident from Figure 5.17 that EBSBM outperforms AEBSBM in this case. The results correspond to the MRR scores, namely 0.8793 with a 95% CI being $(0.8633, 0.8945)$ for EBSBM and 0.7529 for AEBSBM with $(0.7314, 0.7741)$ as a 95% CI. Thus, it is evident from this simulation experiment that a violation of the assumption for the edge-attribute probabilities does impact the nomination performance of the AEBSBM.

According to Amini et al. (2013), spectral clustering techniques in general tend to perform poorly when graphs are sparse. We alleviate this by adding perturbation to the resulting adjacency matrices, $S^{(a)}$, before ASGE as previously discussed in Section 5.3. To further explore how sparsity would affect the nomination performance for AEBSBM, another simulation study is conducted with the values, p, q, and r, being considerably small. Specifically, we set $q_1 = 0.1079, p_1 = 0.0035, r_1 = 0.0020$, and $q_2 = p_2 = r_2 = 0.0164$. Notably, this simulation setting meets all the constraints imposed on the edge-attribute probability vectors. The results show that if the assumptions are satisfied, the sparsity in graphs has no impact on the resulting posterior inference for vertex nomination. For instance, over 1000 graph realizations we obtain a higher MAP score for AEBSBM as compared to EBSBM, specifically 0.7149 with a 95% CI of $(0.6896, 0.7396)$ for AEBSBM and 0.5359 $(0.5109, 0.5609)$ for EBSBM.

The key point from this section is, the AEBSBM will not be as effective as the EBSBM, if all the underlying assumptions for the AEBSBM are not met. Realistically, it is more sensible to compare the nomination performance of the AEBSBM with other models that jointly exploit the content and context information; namely the BVN and C&P models. In this case, the AEBSBM is undoubtedly superior to that of the alternative BVN and C&P models. Particularly, for the Enron graph of week 38, the MAP and MRR scores of C&P are no more than 0.38 and 0.75 respectively (see Coppersmith and Priebe (2012) and Coppersmith (2014) for further details), which are both considerably less than the AEBSBM. Recall that for the same week, we have 0.4414 with a 95% CI being $(0.4189, 0.4641)$ and 0.9666 with $(0.9425, 0.9857)$ as a 95% CI for MAP and MRR, correspondingly.

5.6 Summary

This chapter proposed the AEBSBM as an extension to the vertex nomination model developed in Chapter 4 (EBSBM). Our methodology is motivated by the premise that information relevant to vertex nomination is embedded in both the context and the content, and that utilizing both can potentially lead to an improvement in nomination performance. Thus, in addition to using information from a few observed vertices as in the previous chapter, AEBSBM also leveraged observed edge attributes by exploiting edge-attributed partially observed SBM graphs. Inference with the model was executed via the Metropolis-within-Gibbs algorithm for generating sample points from the posterior distribution.

For ease of comparison the simulation setting of Coppersmith and Priebe (2012) is employed, which has also been used in Chapter 4, to illustrate the nomination performance via the evaluation measures, MAP and MRR. Results from simulation studies show that AEBSBM performs significantly better than chance. In addition, compared with the approach in BVN and EBSBM discussed in Chapter 4, AEBSBM performs increasingly better as the number of unknown red vertices decreases relative to the total number of red vertices. The performance is also elevated as the total number of red vertices grows. In all instances, the proposed model is able to adequately estimate model parameters close to the truth.

An application example is provided using the Enron email corpus where vertices denote employees and their associates, observed red vertices are those proven to have committed fraudulent activity, red edges denote email communications perceived as fraudulent, and where we wish to detect one or more of the remaining unknown vertices as most likely to be fraudsters. Comparisons were made using the MRR and MAP. Based on the MRR score, AEBSBM is comparable to EBSBM. In the case of identifying all red vertices, EBSBM displays a statistically significant superiority in performance over AEBSBM. The violation of the assumptions underpinning AEBSBM is suspected to be the reason for this. This was further investigated in Section 5.5.2 via the simulation study by setting the edge-attributed two-block SBM with parameter values primarily emulating the distribution of content (communication topics) of the Enron graph. The results confirm that violation of the model assumptions can adversely impact the nomination performance of AEBSBM. However, when comparing the results with other alternative joint statistical approaches of Coppersmith and Priebe (2012) and Suwan et al. (2015), AEBSBM gives undeniably a better nomination performance.

Despite the lack of improvement in the performance of AEBSBM when applied to the application example, the simulation results still highlight the essence of vertex nomination where the interplay between edge attributes and graph structure is nontrivial. The superiority in nomination performance of AEBSBM over EBSBM via simulation studies suggests that there exists mutually enriching information in both the context and content. Thus, AEBSBM is worthy of further investigation, for instance examining the circumstances for which jointly harnessing information from content and context is beneficial.

6 | CONCLUDING REMARKS

6.1 Thesis Conclusions

Network data containing information about entities and their interactions are ubiquitous. As the size and complexity of such data increase, the need for innovative ways to process and analyze this data becomes ever more essential. The network modeling literature can be categorized into two broad directions. The first direction focuses on simple and mathematically tractable network models which capture connectivity patterns that exist in real-world networks. Example of such models include the differential attachment model and its variants (Bollobás and Riordan, 2003; Cooper and Frieze, 2003), the small-world model (Watts and Strogatz, 1998), and geometric random graphs (Flaxman et al., 2006). The other direction is statistical modeling, which not only considers the features of vertices and edges in the network, but also the structure of the network (Hoff et al., 2002; Airoldi et al., 2006; Robins et al., 2007a), as discussed in Chapter 2. In many applications an important goal of network modeling is community detection, which helps in the understanding of the structural properties of real-world networks. Although a great deal of effort has been made in discovering the communities in these networks, this task still remains challenging and continues to be of research interest.

This thesis has developed novel empirical Bayes estimation models which employ recently developed theories and results from adjacency spectral embedding of an RDPG for community detection in networks including vertex nomination. Chapter 2 reviewed a number of random graph models for network analysis, such as the Erdös-Rényi graph, the exponential random graph family, SBMs and latent position models. These model various aspects of networks such as homophily, differential attachment, transitivity, and clustering. We discussed the benefits as well as the drawbacks of these models, paying particular attention to latent position models and SBMs. This led us to discuss ASGE and its related works; a technique used in this thesis to estimate the latent positions. A discussion of Bayesian inference as well as the posterior sampling methods, such as Gibbs and Metropolis-Hastings algorithms is also provided.

EBSBM

Chapter 3 formulates an empirical Bayes model for estimating block memberships of vertices in an SBM graph by representing it as an RDPG. Our methodology is motivated by Athreya et al. (2015)'s theoretical developments about the distribution of the adjacency embeddings of RPDGs. Specifically, for an RDPG, they showed that the latent positions estimated via ASGE converge in distribution to a mixture of multivariate normals. As a result, the estimated latent positions of a K-block SBM are independent and identically distributed from a (approximate) K-component multivariate normal mixture. We utilized the theorem of Athreya et al. (2015) to construct our empirical prior distribution for the unknown latent positions. Inference about block membership is conducted by implementing a Metropolis-within-Gibbs algorithm to sample from the posterior distribution.

For comparison purposes, we also formulated an alternative *Flat* and two benchmark models, namely *Exact* and *Gold*, alongside our empirical Bayes model (also dubbed as *ASGE*). These models are named after their respective prior distributions used for the unknown latent positions. The *Exact* model is built as a primary benchmark model since all the model parameters are assumed known except for the block membership vector. A secondary benchmark is the *Gold* model. This is identical to the *Exact* model with an additional latent position parameter that is required to be estimated. The gold standard mixture of normals prior distribution is placed on this parameter which takes its hyperparameters to be the true latent positions and theoretical limiting covariances obtained from the distributional results from Athreya et al. (2015). The *ASGE* model is a key contribution in this thesis, which continues this naming convention as this model employs an empirical prior estimated from the ASGE. Specifically, a $K-$ component multivariate normal mixture is used as a prior on the latent positions. The last model in this chapter, *Flat*, was formulated as an alternative to the *ASGE* model, since without knowledge of the Athreya et al. (2015) theory a natural choice of the prior for the latent positions is a uniform distribution.

The performance of the novel *ASGE* model was demonstrated by three simulation studies and one real data experiment. We first looked at a two-block SBM case where comparisons between the aforementioned models were made. The results showed that the ASGE model consistently outperforms the alternative *Flat* model as well as the GMM. We also illustrated that as the number of vertices increases the probability of mis-assigned vertices significantly declined across all models.

With the re-casting of a K-block SBM as an RDPG, the distribution of the latent positions is a mixture of K point masses since all vertices that belong to the same block will share a common latent vector. To show the robustness of the *ASGE* model, for our second simulation study, we generalized the simulation setting of the two-block SBM where the latent positions are distributed according to a mixture of Dirichlet distributions, instead of a mixture of point masses. Although the performances of the *ASGE* and other alternative models were marginally degraded, the *ASGE* model still comparatively maintained its performance status. This suggests that our *ASGE* model works well in this RDPG generalization of the SBM and is robust to violation of the affinity SBM assumption (having graphs with more connections within blocks and less between).

The last simulation study sought to investigate the performance of the *ASGE* model for a three-block SBM graph whose structure becomes harder to distinguish between blocks. The models were applied on three simulation experiments with slight perturbations to the block probability matrix. That is, the probabilities of an edge existing between a vertex pair within blocks become more similar to one another moving from the first to the third experiment, and keeping the number of vertices fixed where $n = 15$, $n = 150$, and $n = 300$, correspondingly. The rank of the performance between the *ASGE*, *Flat* and GMM is consistent (best to worst) across the three experiments. The models perform statistically significantly better than chance but became progressively worse moving from the first to the third experiment as predicted.

Our methodology was demonstrated using the Wikipedia graph (vertices represent Wikipedia article pages with edges indicating if there is a hyperlink between the associated pages) with comparisons to the aforementioned models. Similar results were evident, specifically, the *ASGE* model illustrated an admirable performance even though this real data set evidently did not meet the affinity assumption placed on the SBM.

EBSBM for vertex nomination

Besides a community detection task in graphs which requires a complete classification of vertices, the notion of identifying only vertices that possess an interesting attribute given a few observed ones (i.e. vertex nomination), has gained ground in recent years. Coppersmith and Priebe (2012) speculated that relevant information for vertex nomination exists in both the graph structure and edge attributes (i.e. context and content, respectively), and that leveraging both via an attributed graph should improve nomination performance. Although inference based on either context or content alone is still possible, which are both explored in Chapters 4 and 5, respectively.

Chapter 4 looked at the extension of the novel empirical Bayes model to address the vertex nomination problem, and focused solely on exploiting the graph structure based on a partially observed SBM graph. Vertex nomination has been cast as a two-block SBM, whereby one of the two blocks is considered to be of interest and only a few block memberships of vertices in this block are observed. A new likelihood function incorporating the additional information from the few observed vertices is constructed together with the previous prior specifications for the model parameters from Chapter 3. Inference about the unknown block memberships of the interesting vertices is then obtained from the resulting posterior distribution. This allows the construction of a ranked nomination list of vertices with unknown block memberships, in which the vertices were ranked in descending order of the posterior probability that a vertex is interesting.

The efficacy of the model was illustrated by two simulation studies; a small-scale, and a large-scale simulated network. Comparisons based on the evaluation measures, mean average precision (MAP) and minimum reciprocal rank (MRR), were made to other known models for vertex nomination, namely those by Coppersmith and Priebe (2012) and Suwan et al. (2015), whose models relied on a collective exploitation of information from attributed graphs compared with that of only exploiting the graph structure employed by the EBSBM. Thus, the competing models were expected to yield a better nomination performance than ours. Results showed that EBSBM was able to comparatively produce as effective as or better nomination performance than its competing models. An example of application is provided using the Enron email corpus dataset, where emails of Enron employees and associates were seized for a legal investigation of financial crimes. The data was made public and has been widely studied for various tasks, vertex nomination in particular. Similar results are evident when the EBSBM was applied to the Enron graph.

AEBSBM for vertex nomination

In Chapter 5, the EBSBM for vertex nomination was further adapted to jointly harness information from both context and content via the exploitation of an attributed graph. For direct comparison, the first part of the simulation study is identical to that of Chapter 4, where the edge-attributed two-block SBM's parameter values for sampling random attributed graphs were from Coppersmith and Priebe (2012) and Suwan et al. (2015). Results show that the AEBSBM provides a statistically significant improvement in the nomination performance over both the non-attributed version (Chapter 4) and the approach in Suwan et al. (2015).

Unexpectedly, when the AEBSBM was applied to the Enron dataset there was no improvement to the nomination performance over its predecessor. The reason for the lack of improvement in the performance for the Enron data was suspected to be the violation of the assumptions underpinning the AEBSBM. To explore this, we conducted a second simulation study in Section 5.5.2 where edge-attributed random graphs were sampled from the two-block SBM with parameter values highlighting the Enron graph structure. The results indeed confirm that if the model assumptions are not fully satisfied, AEBSBM will not improve upon its predecessor (EBSBM). Nonetheless, when comparing the performance of AEBSBM with other alternative joint statistical models, namely Coppersmith and Priebe (2012) and Suwan et al. (2015), AEBSBM yields a substantially better nomination performance. Thus, in spite of the lack of improvement of the AEBSBM on the Enron graph, the simulation studies illuminate the benefits of jointly utilizing both the graph structure and its edge attributes on vertex nomination.

Although community detection in graphs has been one of many important problems since the emergence of the graphical representation of relational data, vertex nomination is relatively new and with ample avenues of further research. Overall, this thesis has introduced the empirical Bayes model for community detection and vertex nomination. Simulation studies and real-world applications have shown that the models are effective tools, providing promising solutions within the statistical network modeling literature.

6.2 Discussion of Future Research

There are many future directions for this research to be considered. The aim of this thesis has been to demonstrate the utility of a multivariate Gaussian mixture, estimated through adjacency spectral embedding as an empirical prior distribution in a Bayesian inference methodology for block membership estimation and vertex nomination in an SBM graph. This thesis focused on simple undirected graphs with no self-loops; extension to directed and weighted graphs with moderate modifications to the novel methodology is of both theoretical and practical interest.

A further area of potential new research is that of model selection techniques which is in general a difficult problem. Automatic determination of both the dimension, d, for a truncated eigendecomposition and the complexity, K, for a Gaussian mixture model estimate are significant practical problems and have garnered much attention in both the applied and theoretical literature. For our case, Fishkind et al. (2013b) show that the SBM embedding dimension, d, can be successfully estimated, and Fraley and Raftery (2002) give a common approach to estimating the number of Gaussian mixture components, K. In order to avoid this dilemma, we simply assumed that the number of blocks, K, and the dimension of the latent positions, d, are known. In addition to knowing d and K, we also assumed that $d = K$. This choice is justified for the adjacency spectral embedding dimension of an SBM, as increasing d beyond the true latent position dimension adds variance without a concomitant reduction in bias. It may be productive to investigate simultaneous model selection methodologies for d and K. Moreover, the robustness of the empirical Bayes model to misspecification of d and K is also of great practical importance.

We mainly considered presenting results in the dense/non-sparse regime (where the number of edges is close to the maximal number of edges) in which raw spectral embedding, even without the empirical Bayes augmentation, can provide strongly consistent classification and clustering (Lyzinski et al., 2013; Sussman et al., 2012a). However, as pointed out in Chapter 3 this does not rule out the possibility of substantial performance gains for finite sample sizes. It is the finite sample performance gains that are the main topic of this research. While Sussman (2014) provides a non-dense version of Athreya et al. (2015)'s CLT, both theoretical and methodological issues remain in developing its utility for generating an empirical prior. This is of considerable interest and thus a more comprehensive understanding of the CLT for non-dense RDPGs is a priority for ongoing research.

Moreover, throughout this thesis, we have only looked at the SBM which generates communities with higher edge densities within blocks and less between blocks (i.e. an affinity SBM) to circumvent the non-identifiability issue. But of course in many real-world applications this might not always be true, thus generalizing the empirical Bayes model for other SBM graph structures including hierarchical, core-periphery, or structures that possess "hubs" or high-degree vertices could be another venue for future work. For instance, further extensions of the empirical Bayes model for the degree-corrected SBM of Karrer and Newman (2011) that allow for heterogeneous degrees, the weighted SBM of (Aicher et al., 2014) where edges can have weight attached to them, or the hierarchical SBM of (Lyzinski et al., 2015) that enriches a natural hierarchical structure. With such graph structures this may provoke a label-switching problem in the MCMC output of the empirical Bayes model, which needs to be dealt with cautiously.

Chapters 4 and 5 extended the model to perform vertex nomination. The experiments presented place a great emphasis on identifying only interesting vertices based on their marginal posterior probabilities of being interesting. However, in reality, it may be of practical imperative to identify two or more jointly interesting vertices. For example, in the context of fraudulent activity in a company, one may wish to find a fraudster together with any accomplices. Since we have MCMC sample points from the full joint posterior distribution, it is reasonably straightforward to obtain the required posterior joint probabilities. Our methods can also easily be generalized to situations where there are more than two vertex attributes, as opposed to identifying vertices with just the attribute of interest. Furthermore, for this exploitation task, we have fixed the number of blocks, $K = 2$; one of which is of interest, however, this restriction can be dropped. It is relatively straightforward to modify our vertex nomination models to have $K \geq 2$ such that one of the blocks consists of interesting vertices.

In Chapter 5, the empirical Bayes model was further extended to incorporate edge-attribute information derived from attributed graphs for vertex nomination. Here, we have assumed that the edges and their attributes are perfectly observed. In real life applications, some errors may arise in the attribution process leading to missing edges and missing edge-attributes. Thus, extension to handling missing data is another avenue future work. This can possibly be achieved following a similar approach to Aicher et al. (2014) by making a distinction between (i) no edge presence (observed absence of a tie between a vertex pair), (ii) missing edge (an unobserved tie between a vertex pair), and (iii) a missing edge attribute (observed tie but unobserved edge attribute between a vertex pair). These explicit differences can be integrated as part of the model formulation to tackle imperfectly observed data.

The superiority in the nomination performance of the AEBSBM over its predecessor via simulation studies indicates that there is complementary information in both the context and the content despite the lack of performance improvement from the application to the Enron data in Chapter 5. Thus, the AEBSBM is worthy of further development. This may include investigating the situations for which jointly exploiting information from content and context is beneficial, and the assumptions for vertex nomination that lead to less restrictive constraints for the model parameters. Also, the proposed models in this thesis are concerned with analyzing a single snapshot of the network. However in many applications multiple snapshots of random graphs may be accessible (a time series of graphs), thus generalizing the model by utilizing this additional information can potentially improve the nomination performance (Priebe et al., 2005).

References

Aicher, C., Jacobs, A. Z., and Clauset, A. (2014), "Learning Latent Block Structure in Weighted Networks," *arXiv preprint arXiv:1404.0431*. 25, 39, 109, 113, 146, 147

Airoldi, E., Blei, D., Fienberg, S., and Xing, E. (2008), "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research.*, 9, 1981–2014. 7, 24, 39

Airoldi, E. M., Blei, D. M., Fienberg, S. E., Xing, E. P., and Jaakkola, T. (2006), "Mixed membership stochastic block models for relational data with application to protein-protein interactions," in *Proceedings of the international biometrics society annual meeting*, pp. 1–34. 111, 141

Airoldi, E. M., Costa, T. B., and Chan, S. H. (2013), "Stochastic blockmodel approximation of a graphon: Theory and consistent estimation," in *Advances in Neural Information Processing Systems*, pp. 692–700. 80

Allman, E. S., Matias, C., and Rhodes, J. A. (2011), "Parameter identifiability in a class of random graph mixture models," *Journal of Statistical Planning and Inference*, 141, 1719–1736. 23

Amini, A. A., Chen, A., Bickel, P. J., Levina, E., et al. (2013), "Pseudo-likelihood methods for community detection in large sparse networks," *The Annals of Statistics*, 41, 2097–2122. 118, 137

Anderson, C., Wasserman, S., and Faust, K. (1992), "Building stochastic blockmodels," *Social Networks*, 14, 137 – 161. 23, 24

Athreya, A., Priebe, C., Tang, M., Lyzinski, V., Marchette, D., and Sussman, D. (2015), "A Limit Theorem for Scaled Eigenvectors of Random Dot Product Graphs," *Sankhya A*, 1–18. 1, 4, 6, 7, 9, 25, 29, 30, 31, 40, 43, 45, 86, 107, 142, 146

Banfield, J. D. and Raftery, A. E. (1993), "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, 803–821. 32

Bastian, M., Heymann, S., Jacomy, M., et al. (2009), "Gephi: an open source software for exploring and manipulating networks." *ICWSM*, 8, 361–362. vi, 92, 105

Bell, R., Koren, Y., and Volinsky, C. (2008), "The bellkor 2008 solution to the netflix prize," *Statistics Research Department at AT&T Research*. 4, 84

Bernardo, J. M. and Smith, A. F. (2009), *Bayesian Theory*, vol. 405, John Wiley & Sons. 33

Berry, M. W., Browne, M., and Signer, B. (2001), "topic annotated Enron email data set," *Philadelphia: Linguistic Data Consortium*. 130

Bethard, S. and Jurafsky, D. (2010), "Who should I cite: learning literature search models from citation behavior," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ACM, pp. 609–618. 109, 111

Bickel, P. and Chen, A. (2009), "A nonparametric view of network models and Newman–Girvan and other modularities," *Proceedings of the National Academy of Sciences*, 106, 21068–21073. 6, 23, 24, 39

Bickel, P., Choi, D., Chang, X., and Zhang, H. (2013), "Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels," *The Annals of Statistics*, 41, 1922–1943. 6, 39

Bishop, C. (2007), "Pattern Recognition and Machine Learning (Information Science and Statistics)," . 82

Bollobás, B. (2001), *Random graphs*, vol. 73, Cambridge University Press. 111

Bollobás, B. and Riordan, O. M. (2003), "Mathematical results on scale-free random graphs," *Handbook of Graphs and Networks: from the genome to the internet*, 1–34. 141

Borges, N., Coppersmith, G., Meyer, G. G., Priebe, C. E., et al. (2011), "Anomaly detection for random graphs using distributions of vertex invariants," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, IEEE, pp. 1–6. 111

Botev, Z., Grotowski, J., and Kroese, D. (2010), "Kernel density estimation via diffusion," *The Annals of Statistics*, 38, 2916–2957. 61

Brinda, W., Jain, S., and Trosset, M. (2011), "Inference on random graphs with classified edge attributes," Tech. rep., Technical Report 11-03, Department of Statistics, Indiana University. 112

Brooks, S. P. and Roberts, G. O. (1998), "Convergence assessment techniques for Markov chain Monte Carlo," *Statistics and Computing*, 8, 319–335. 56

Caimo, A. and Friel, N. (2011), "Bayesian inference for exponential random graph models," *Social Networks*, 33, 41–55. 21

Carlin, B. P. and Louis, T. A. (2011), *Bayesian methods for data analysis*, CRC Press. 34

Carrington, P. J., Scott, J., and Wasserman, S. (2005), *Models and methods in social network analysis*, vol. 28, Cambridge University Press. 22

Celeux, G. (1998), "Bayesian inference for mixture: The label switching problem," in *Compstat*, Springer, pp. 227–232. 53

Celeux, G., Hurn, M., and Robert, C. P. (2000), "Computational and inferential difficulties with mixture posterior distributions," *Journal of the American Statistical Association*, 95, 957–970. 53

Celisse, A., Daudin, J.-J., and Pierre, L. (2012), "Consistency of maximum-likelihood and variational estimators in the stochastic block model," *Electronic Journal of Statistics*, 6, 1847–1899. 6, 39

Chandola, V., Banerjee, A., and Kumar, V. (2009), "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, 41, 15. 84

Choi, D. S., Wolfe, P. J., and Airoldi, E. M. (2012), "Stochastic blockmodels with a growing number of classes," *Biometrika*, 99, 273–284. 6, 24, 39

Clauset, A., Moore, C., and Newman, M. E. (2008), "Hierarchical structure and the prediction of missing links in networks," *Nature*, 453, 98–101. 23

Cooper, C. and Frieze, A. (2003), "A general model of web graphs," *Random Structures & Algorithms*, 22, 311–335. 141

Coppersmith, G. (2014), "Vertex nomination," *Wiley Interdisciplinary Reviews: Computational Statistics*, 6, 144–153. 5, 81, 84, 108, 136, 137

Coppersmith, G. and Priebe, C. (2012), "Vertex nomination via content and context," *arXiv:1201.4118v1*. ii, 5, 10, 11, 81, 82, 83, 90, 91, 108, 110, 113, 114, 116, 117, 122, 130, 137, 138, 143, 144, 145

Cortes, C., Pregibon, D., and Volinsky, C. (2001), "Communities of Interest," in *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, London, UK, UK: Springer-Verlag, IDA'01, pp. 105–114. 4

Cowles, M. K. and Carlin, B. P. (1996), "Markov chain Monte Carlo convergence diagnostics: A comparative review," *Journal of the American Statistical Association*, 91, 883–904. 56

Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011), "Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications," *Physical Review E*, 84, 066106. 47

Duijn, M. A., Snijders, T. A., and Zijlstra, B. J. (2004), "p_2: a random effects model with covariates for directed graphs," *Statistica Neerlandica*, 58, 234–254. 19, 20

Efron, B. (1987), "Better bootstrap confidence intervals," *Journal of the American Statistical Association*, 82, 171–185. vi, 57, 64

Eldardiry, H. and Neville, J. (2012), "An analysis of how ensembles of collective classifiers improve predictions in graphs," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, ACM, pp. 225–234. 111

Erdös, P. and Rényi, A. (1959), "On random graphs I." *Publications Mathematicae Debrecen*, 6, 290–297. 2, 17

Fienberg, S. E. and Wasserman, S. S. (1981), "Categorical data analysis of single sociometric relations," *Sociological Methodology*, 156–192. 19

Fishkind, D. E., Lyzinski, V., Pao, H., Chen, L., and Priebe, C. E. (2013a), "Vertex Nomination Schemes for Membership Prediction," *arXiv preprint arXiv:1312.2638*. 6, 7, 24, 29, 83

Fishkind, D. E., Sussman, D. L., Tang, M., Vogelstein, J. T., and Priebe, C. E. (2013b), "Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown," *SIAM Journal on Matrix Analysis and Applications*, 34, 23–39. 6, 23, 39, 75, 79, 146

Flake, G. W., Lawrence, S., Giles, C. L., and Coetzee, F. M. (2002), "Self–organization and identification of web communities," *Computer*, 35, 66–70. 2

Flaxman, A. D., Frieze, A. M., and Vera, J. (2006), "A geometric preferential attachment model of networks," *Internet Mathematics*, 3, 187–205. 141

Fortunato, S. (2010), "Community detection in graphs," *Physics Reports*, 486, 75–174. 3, 5, 16

Fraley, C. and Raftery, A. E. (1999), "MCLUST: Software for model-based cluster analysis," *Journal of Classification*, 16, 297–306. 31

— (2002), "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American statistical Association*, 97, 611–631. 32, 44, 79, 146

Frank, O. and Strauss, D. (1986), "Markov graphs," *Journal of the American Statistical Association*, 81, 832–842. 20

Friel, N., Ryan, C., and Wyse, J. (2013), "Bayesian model selection for the latent position cluster model for Social Networks," *arXiv preprint arXiv:1308.4871*. 8, 28

Gelfand, A. E. and Smith, A. F. (1990), "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, 85, 398–409. 34, 37

Gelman, A. (1996), "Inference and monitoring convergence," in *Markov chain Monte Carlo in practice*, Springer, pp. 131–143. 56, 57

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014), *Bayesian data analysis*, vol. 2, Taylor & Francis. 34

Gelman, A. and Rubin, D. B. (1992), "Inference from iterative simulation using multiple sequences," *Statistical Science*, 457–472. 56, 57, 91

Gelman, A. and Shirley, K. (2011), "Inference from simulations and monitoring convergence," *Handbook of Markov chain Monte Carlo*, 163–174. 55

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 721–741. 34, 36

Geweke, J. (1989), "Bayesian inference in econometric models using Monte Carlo integration," *Econometrica: Journal of the Econometric Society*, 1317–1339. 34

Ghosh, P., Gill, P., Muthukumarana, S., and Swartz, T. (2010), "A semiparametric Bayesian approach to network modelling using Dirichlet process prior distributions," *Australian & New Zealand Journal of Statistics*, 52, 289–302. 16

Gill, P. and Swartz, T. (2004), "Bayesian analysis of directed graphs data with applications to social networks," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53, 249–260. 7, 20

Girvan, M. and Newman, M. E. (2002), "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, 99, 7821–7826. 2, 5

Goldenberg, A., Zheng, A., Fienberg, S., and Airoldi, E. (2009), "A survey of statistical network models." *Foundations & Trends in Machine Learning*, 2, 129 – 233. 3, 16, 17, 18

Gorin, A., Priebe, C., and Grothendieck, J. (2010), "Random attributed graphs for statistical inference from content and context," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, IEEE, pp. 5430–5433. 27, 112

Grothendieck, J., Priebe, C., and Gorin, A. (2010), "Statistical inference on attributed random graphs: Fusion of graph features and content," *Computational Statistics & Data Analysis*, 54, 1777–1790. 5, 84, 111

Handcock, M., Raftery, A., and Tantrum, J. (2007), "Model-based clustering for social networks," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 170, pp. 301–354. 8, 24, 27, 28, 39, 44

Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, 97–109. 34, 35, 36

Hoff, P., Raftery, A., and Handcock, M. (2002), "Latent space approaches to social network analysis," *Journal of the american Statistical association*, 97, 1090–1098. 3, 6, 8, 20, 25, 26, 27, 39, 44, 114, 141

Holland, P. and Leinhardt, S. (1981), "An exponential family of probability distributions for directed graphs," *Journal of the American Statistical Association*, 76, 33–50. 7, 18, 19, 23

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983), "Stochastic blockmodels: First steps," *Social Networks*, 5, 109–137. 3, 22, 109

Huang, Z., Li, X., and Chen, H. (2005), "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, ACM, pp. 141–142. 111

Hurn, M., Justel, A., and Robert, C. P. (2003), "Estimating mixtures of regressions," *Journal of Computational and Graphical Statistics*, 12, 55–79. 53

Jasra, A., Holmes, C., and Stephens, D. (2005), "Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling," *Statistical Science*, 50–67. 53

Jordan, M. (2004), "Graphical models," *Statistical Science*, 140–155. 48

Karrer, B. and Newman, M. E. (2011), "Stochastic blockmodels and community structure in networks," *Physical Review E*, 83, 016107. 24, 146

Latouche, P., Birmele, E., and Ambroise, C. (2012), "Variational Bayesian inference and complexity control for stochastic block models," *Statistical Modelling*, 12, 93–115. 24

Lee, N., Leung, T., and Priebe, C. (2011), "Random Graphs Based on Self–Exciting Messaging Activities," Unpublished. 83, 108, 114

Li, J. and Zaïane, O. R. (2004), “Combining usage, content, and structure data to improve web site recommendation,” in *E-Commerce and Web Technologies*, Springer, pp. 305–315. 111

Lyzinski, V., Sussman, D., Tang, M., Athreya, A., and Priebe, C. (2013), “Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding,” *arXiv preprint arXiv:1310.0532*. 6, 29, 31, 79, 146

Lyzinski, V., Sussman, D. L., Fishkind, D. E., Pao, H., Chen, L., Vogelstein, J. T., Park, Y., and Priebe, C. E. (2014), “Spectral Clustering for Divide-and-Conquer Graph Matching,” *stat*, 1050, 22. 7

Lyzinski, V., Tang, M., Athreya, A., Park, Y., and Priebe, C. E. (2015), “Community Detection and Classification in Hierarchical Stochastic Blockmodels,” *arXiv preprint arXiv:1503.02115*. 146

Manning, C. D., Raghavan, P., and Schütze, H. (2008), *Introduction to Information Retrieval*, vol. 1, Cambridge University Press Cambridge. 90

Marchette, D., Priebe, C., and Coppersmith, G. (2011), “Vertex nomination via attributed random dot product graphs,” in *Proceedings of the 57th ISI World Statistics Congress*, vol. 6, p. 16. 83, 114, 115, 116, 118

Marchette, D. J. and Priebe, C. E. (2008), “Predicting unobserved links in incompletely observed networks,” *Computational Statistics & Data Analysis*, 52, 1373–1386. 111

MATLAB (2015), *version 7.10.0 (R2015a)*, Natick, Massachusetts: The MathWorks Inc. 60

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), “Equation of state calculations by fast computing machines,” *The Journal Of Chemical Physics*, 21, 1087. 34, 35

Newman, M. E. (2004), “Detecting community structure in networks,” *The European Physical Journal B-Condensed Matter and Complex Systems*, 38, 321–330. 5

— (2006), “Modularity and community structure in networks,” *Proceedings of the National Academy of Sciences*, 103, 8577–8582. 6, 39

Newman, M. E. and Girvan, M. (2004), “Finding and evaluating community structure in networks,” *Physical review E*, 69, 026113. 6

Nickel, C. (2006), “Random dot product graphs: A model for social networks,” Ph.D. thesis, Johns Hopkins University. 3, 6, 28

Nowicki, K. and Snijders, T. (2001), “Estimation and prediction for stochastic blockstructures,” *Journal of the American Statistical Association*, 96, pp. 1077–1087. 7, 24, 39, 42, 112

Olhede, S. C. and Wolfe, P. J. (2013), “Network histograms and universality of blockmodel approximation,” *arXiv preprint arXiv:1312.5306*. 80

Ouzienko, V. and Guo, Y.and Obradovic, Z. (2011), “A decoupled exponential random graph model for prediction of structure and attributes in temporal social networks,” *Statistical Analysis and Data Mining*, 4, 470–486. 22

Pao, H., Coppersmith, G. A., and Priebe, C. E. (2011), "Statistical inference on random graphs: Comparative power analyses via Monte Carlo," *Journal of Computational and Graphical Statistics*, 20, 395–416. 111

Park, Y., Moore, C., and Bader, J. S. (2010), "Dynamic networks from hierarchical Bayesian graph clustering," *PloS one*, 5, e8118. 23

Pavlov, D. Y. and Pennock, D. M. (2002), "A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains," in *Advances in neural information processing systems*, pp. 1441–1448. 111

Pearl, J. (2014), *Probabilistic reasoning in intelligent systems: Networks of plausible inference*, Morgan Kaufmann. 48

Priebe, C., Conroy, J., Marchette, D., and Park, Y. (2005), "Scan statistics on Enron graphs," *Computational & Mathematical Organization Theory*, 11, 229–247. 84, 104, 111, 130, 147

Priebe, C., Park, Y., Marchette, D., Conroy, J., Grothendieck, J., and Gorin, A. (2010), "Statistical inference on attributed random graphs: Fusion of graph features and content: An experiment on time series of Enron graphs," *Computational Statistics* & *Data Analysis*, 54, 1766 – 1776. 84, 112

Qi, G., Aggarwal, C., Qi, T., Ji, H., and Huang, T. (2012a), "Exploring Context and Content Links in Social Media: A Latent Space Method," *IEEE Trans. Pattern Anal. Mach. Intell.*, 34, 850–862. 83

Qi, G.-J., Aggarwal, C., Tian, Q., Ji, H., and Huang, T. S. (2012b), "Exploring context and content links in social media: A latent space method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34, 850–862. 114

Resnick, P. and Varian, H. R. (1997), "Recommender systems," *Communications of the ACM*, 40, 56–58. 3, 4, 82, 84

Richardson, S. and Green, P. J. (1997), "On Bayesian analysis of mixtures with an unknown number of components (with discussion)," *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59, 731–792. 53

Rives, A. W. and Galitski, T. (2003), "Modular organization of cellular networks," *Proceedings of the National Academy of Sciences*, 100, 1128–1133. 2

Robins, G., Pattison, P., Kalish, Y., and Lusher, D. (2007a), "An introduction to exponential random graph (p^*) models for social networks," *Social Networks*, 29, 173 – 191. 141

Robins, G., Snijders, T., Wang, P., Handcock, M., and Pattison, P. (2007b), "Recent developments in exponential random graph (p^*) models for social networks," *Social Networks*, 29, 192 – 215. 21

Rodríguez, A. (2012), "Modeling the dynamics of social networks using Bayesian hierarchical blockmodels," *Statistical Analysis and Data Mining*, 5, 218–234. 8, 24

Rohe, K., Chatterjee, S., and Yu, B. (2011), “Spectral clustering and the high-dimensional stochastic blockmodel,” *The Annals of Statistics*, 39, 1878–1915. 5, 24, 39

Salter-Townshend, M. and Murphy, T. B. (2013), “Variational Bayesian inference for the latent position cluster model for network data,” *Computational Statistics & Data Analysis*, 57, 661–671. 27

Salter-Townshend, M., White, A., Gollini, I., and Murphy, T. (2012), “Review of statistical network analysis: Models, algorithms, and software,” *Statistical Analysis and Data Mining*, 5, 243–264. 18, 25

Scheinerman, E. R. and Tucker, K. (2010), “Modeling graphs using dot product representations,” *Computational Statistics*, 25, 1–16. 114

Shawe-Taylor, J. and Cristianini, N. (2004), *Kernel methods for pattern analysis*, Cambridge University Press. 115

Smith, A. F. and Gelfand, A. E. (1992), “Bayesian statistics without tears: a sampling–resampling perspective,” *The American Statistician*, 46, 84–88. 34

Snijders, T. (2002), “Markov chain Monte Carlo estimation of exponential random graph models,” *Journal of Social Structure*, 3, 1–40. 21

Snijders, T. and Nowicki, K. (1997), “Estimation and prediction for stochastic blockmodels for graphs with latent block structure,” *Journal of Classification*, 14, 75–100. 7, 24, 39, 42, 112

Spirin, V. and Mirny, L. A. (2003), “Protein complexes and functional modules in molecular networks,” *Proceedings of the National Academy of Sciences*, 100, 12123–12128. 2

Steinley, D., Brusco, M., and Wasserman, S. (2011), “Clusterwise p^* models for social network analysis,” *Statistical Analysis and Data Mining*, 4, 487–496. 21

Steinley, D. and Wasserman, S. (2011), “Introduction: Special issue of statistical analysis and data mining on networks,” *Statistical Analysis and Data Mining*, 4, 459–460. 22

Stephens, M. (2000), “Dealing with label switching in mixture models,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62, 795–809. 53

Stephens, M. and Phil, D. (1997), “Bayesian methods for mixtures of normal distributions,” . 53

Strauss, D. and Ikeda, M. (1990), “Pseudo-likelihood estimation for social networks,” *Journal of the American Statistical Association*, 85, 204–212. 21

Sun, M., Tang, M., and Priebe, C. (2012), “A Comparison of Graph Embedding Methods for Vertex Nomination,” in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, IEEE, vol. 1, pp. 398–403. 83

Sussman, D. L. (2014), “Foundations of Adjacency Spectral Embedding,” Ph.D. thesis, Johns Hopkins University. 30, 65, 79, 146

Sussman, D. L., Tang, M., Fishkind, D. E., and Priebe, C. E. (2012a), “A consistent adjacency spectral embedding for stochastic blockmodel graphs,” *Journal of the American Statistical Association*, 107, 1119–1128. 3, 6, 7, 23, 24, 29, 30, 39, 40, 79, 146

Sussman, D. L., Tang, M., and Priebe, C. E. (2012b), "Universally Consistent Latent Position Estimation and Vertex Classification for Random Dot Product Graphs," *arXiv preprint arXiv:1207.6745*. 23, 29

— (2014), "Consistent latent position estimation and vertex classification for random dot product graphs," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 48–57. 23, 31

Suwan, S., Lee, D. S., and Priebe, C. E. (2015), "Bayesian Vertex Nomination Using Content and Context," *Wiley Interdisciplinary Reviews: Computational Statistics*, 7, 400–416. ii, vii, 5, 10, 11, 12, 83, 91, 97, 108, 116, 117, 122, 130, 138, 144, 145

Tang, M., Athreya, A., Sussman, D. L., Lyzinski, V., and Priebe, C. E. (2014), "A nonparametric two-sample hypothesis testing problem for random dot product graphs," *arXiv preprint arXiv:1409.2344*. 7, 23, 29

Tang, M., Park, Y., Lee, N. H., and Priebe, C. E. (2013a), "Attribute fusion in a latent process model for time series of graphs," *Signal Processing, IEEE Transactions on*, 61, 1721–1732. 112

Tang, M., Park, Y., and Priebe, C. E. (2013b), "Out-of-sample extension for latent position graphs," *arXiv preprint arXiv:1305.4893*. 7, 31

Tang, M., Sussman, D. L., Priebe, C. E., et al. (2013c), "Universally consistent vertex classification for latent positions graphs," *The Annals of Statistics*, 41, 1406–1430. 7, 79, 84

Tjelmeland, H. and Besag, J. (2001), "Markov random fields with higher–order interactions," *Scandinavian Journal of Statistics*, 25, 415–433. 21

van Duijn, M. A., Gile, K. J., and Handcock, M. S. (2009), "A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models," *Social Networks*, 31, 52–62. 21

Vogelstein, J. T., Roncal, W. G., Vogelstein, R. J., and Priebe, C. E. (2013), "Graph classification using signal-subgraphs: Applications in statistical connectomics," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35, 1539–1551. 111

Wang, Y. J. and Wong, G. Y. (1987), "Stochastic blockmodels for directed graphs," *Journal of the American Statistical Association*, 82, 8–19. 20, 23

Wasserman, S. (1994), *Social network analysis: Methods and applications*, vol. 8, Cambridge University Press. 23

Wasserman, S. and Anderson, C. (1987), "Stochastic a posteriori blockmodels: Construction and assessment," *Social Networks*, 9, 1–36. 22, 24

Wasserman, S. and Pattison, P. (1996), "Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p," *Psychometrika*, 61, 401–425. 21

Watts, D. J. and Strogatz, S. H. (1998), "Collective dynamics of small-world networks," *nature*, 393, 440–442. 141

Wong, G. Y. (1987), "Bayesian models for directed graphs," *Journal of the American Statistical Association*, 82, 140–148. 7

Young, S. J. and Scheinerman, E. R. (2007), “Random dot product graph models for social networks,” in *Algorithms and models for the web-graph*, Springer, pp. 138–149. 3, 6, 28, 40, 115

Zhao, Y., Levina, E., Zhu, J., et al. (2012), “Consistency of community detection in networks under degree–corrected stochastic block models,” *The Annals of Statistics*, 40, 2266–2292. 24

Zijlstra, B. J. H., van Duijn, M. A., and Snijders, T. A. (2006), “The multilevel p_2 model,” *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, 2, 42–47. 20