Selecting Generative Models for Networks using Classification with Machine Learning

Nicholas Alfredo Larsen

An undergraduate senior honors thesis submitted to the faculty of the University of North Carolina at Chapel Hill in fulfillment of the requirements for the Honors Carolina Senior Thesis in the Department of Statistics and Operations Research.

Chapel Hill 2018

To be approved by: Peter J. Mucha Nicolas Fraiman Shankar Bhamidi

©2018 Nicholas A. Larsen ALL RIGHTS RESERVED

ABSTRACT

NICHOLAS A. LARSEN: Model Selection with Machine Learning for Classifying Network Community Structures (Under the direction of Peter J. Mucha, Shankar Bhamidi, and Nicolas Fraiman)

By representing data entities as a map of edges and vertices, where each edge encodes a relationship between two vertices, networks have an almost unlimited ability to capture relationships and patterns impossible to see with the human eye. Because these patterns often reflect key aspects of the data, a significant portion of network science is devoted to detecting and distinguishing networks by using these topological features. The use of machine learning for classifying networks is a popular solution; research in this area includes techniques ranging from k-Nearest Neighbors to language modeling-inspired deep learning methods. Another area of interest with respect to networks is model selection, which can provide unique insights into a graph's topological and probabilistic properties. This thesis combines the two areas of network classification with machine learning and generative model selection by using the popular algorithm known as "random forests" as a potential model selection criterion. First, we perform a series of experiments designed to characterize the discriminatory power of random forests on a wide variety of synthetic graphs generated by dozens of Stochastic Block Models (SBMs). Then, we take advantage of well-known network structural properties and compare the generative model of best fit selected by random forests to the model chosen by a previously established selection criterion known as Integrated Completed Likelihood (ICL). In applying these techniques to selecting Erdős-Rényi mixture models for a macaque brain connectivity dataset and using the model that maximizes the ICL criterion as the "gold standard," we observed that random forests serves as a comparable model selection method when using topological network statistics as the feature space, selecting the same best-fit model chosen by ICL over 95% of the time.

To my mentors, Peter Mucha and Natalie Stanley, who became my academic family.

ACKNOWLEDGEMENTS

It goes without saying that I would acknowledge my primary advisor, Peter Mucha, for his advice and direction in implementing this project. What is, perhaps, less obvious is my sincere gratitude for having a role-model who goes above and beyond his role as an academic advisor. To this day I am not sure why he took on a freshman undergraduate with no coding experience, but I will always be grateful he did. I am truly honored to have benefited from Peter's mentorship; our subsequent weekly meetings have been an invaluable experience and have included some of the most stimulating conversations during my time at Carolina. I have learned so much beyond network science - from the realities of administration and research in the "real world" to what pro time management looks like - that listing them under the Acknowledgements section in an undergraduate thesis would be a bit ridiculous. Thank you for always making me feel welcome.

Behind any undergraduate researcher, there is a graduate student who actually knows what the project is about. I would like to thank Natalie Stanley not only for her friendship, but also for showing me what true persistence and initiative looks like. Yours is the standard that I have set for myself as I enter graduate school.

Many thanks to the Mucha research group, whose meetings have served as a collective rolemodel and became a weekly highlight. I would also like to thank Shankar Bhamidi, who despite an extraordinarily busy year sponsored an extra undergraduate student and put so much effort into writing recommendation letters and providing useful insights in my search for graduate schools. Thank you also to Nicolas Fraiman, who got on board with such short notice.

To my parents, Susana and Rick: there's not much I can say that you don't already know - thank you for your unconditional support and for remembering my core. To my older sister and roommate, Alexandra, who has given me an invaluable window into the future. I would also like to thank Allison Wang, who makes sure I remember how to be distracted every now and then. I can't imagine my life at UNC without you!

TABLE OF CONTENTS

LI	ST OF	TABL	ES	viii
LI	ST OF	FIGUR	RES	xi
LI	ST OF	ABBR	EVIATIONS	xiii
1	Intro	duction		1
	1.1	Overvi	ew of Network Classification and Modeling	2
	1.2	Motiva	tion and Goals	3
2	Expe	riments	with Synthetic data sets	5
	2.1	Introdu	ction to Stochastic Block Models	6
	2.2	Motiva	tion for Synthetic Experiments and Description of SBM Parameters	8
	2.3	Experin	ments Using Flattened Adjacency Matrices as a Feature Space	9
		2.3.1	Using flattened adjacency matrices as feature sets	9
		2.3.2	Classification of graphs of same type, where only the number of communities differ.	11
		2.3.3	Classify graphs of the same type, where each label corresponds to a probability matrix with different parameter values	12
		2.3.4	Summary of results on the flattened adjacency feature space	15
	2.4	Using I	Network Statistics	15
		2.4.1	Description of network statistics	17
	2.5	Experin	ments Using Network Statistics as a Feature Space	19
		2.5.1	Classification of graphs of same type, where only the number of communities differ.	19
		2.5.2	Classification graphs of the same type, where each label corre- sponds to a probability matrix with different parameter values	22

	2.6	Summary		22	
3	Mod	del Selection Using Random Forests			
	3.1	Overvi	ew of the <i>R</i> package 'mixer' and data set	26	
		3.1.1	Brief overview of MixNet models	27	
		3.1.2	MixNet model estimation	28	
	3.2	Model	Selection using Random Forests	29	
		3.2.1	Model selection with edge-based classification	30	
		3.2.2	Model selection with network statistics-based classification	31	
		3.2.3	Results	31	
4	Cond	clusion a	and Future Directions	34	
BI	BIBLIOGRAPHY				

LIST OF TABLES

2.1	Summary of Total Number of SBMs and Instance Graphs. For each SBM, we generated 100 instance graphs for use in evaluating the discriminatory power over these sets of SBM parameters.	9
2.2	Summary of SBM parameters used for classification of graphs of same type, where only the number of communities differ. Networks generated by these models were used in sections 2.3.2 and 2.5.1. n is the number of nodes for each graph, p_{ij} represents the (i, j) th entry in the probability matrices. In the case of the core-periphery structure, λ is the rate of exponential decay as defined earlier. The parameters summarized here represent total of 15 SBMs with three models per graph type, one for each k .	9
2.3	Summary of SBM parameters used for classification of graphs of the same type, where each label corresponds to a probability matrix with different parameter values. See tables 2.4, 2.5, and 2.6 for summary of the core-periphery structure parameters. Networks generated by these models were used in sections 2.3.3 and 2.5.2. For all graph types we have fixed the number of communities at 5. In the case of the random graphs, values of p have been chosen to generate graphs with a range of densities. Smaller values of p indicate low edge probability, resulting in sparser networks. Conversely, as p increases, the network density grows as well. For assortative graphs, we have fixed inter-community probability as 0.5 and vary the between-community edge probabilities (p_{ij} where $j \neq i$). As p_{ij} increases, the assortative structure more closely resembles that of a random graph. The disassortative parameters mimic the assortative case, in that we only vary p_{ij} . Values of p_{ij} are selected such that nodes in graphs from differing labels are twice as likely or half as likely to connect to nodes in other communities. Other parameters for the disassortative graphs are selected such that two labels have the same value for $p_{ii} - p_{i,i\pm 1}$ and other labels are slightly sparser or denser versions	10
2.4	Parameter Summary for Core-periphery Case, Sub-Experiment 1. (Reference <i>item 5</i> in section 2.1 for notation.) Networks generated by these models were used in sections 2.3.3 and 2.5.2. Labels 0, 1, and 2 correspond to networks with strong inner-core edge probability (P_0) with varying rates of decay (λ). Labels 3, 4, and 5 test the same idea with a weaker inner-core edge probability. Labels 6 and 7 were used to test classification problems where networks have the same inner-core edge probabilities and similar rates of decay. They were tested with labels 0 and 3 respectively.	10

2.5	Parameter Summary for Core-periphery Case, Sub-Experiment 2 . Labels 8 and 9 were used for testing discriminatory power in the case of similar cores and decay rates when the networks are dense. Labels 10 and 3 were used for the same purpose except for sparse networks. Networks generated by these models were used in sections 2.3.3 and 2.5.2	10
2.6	Parameter Summary for Core-periphery Case, Sub-Experiment 3 . Labels 0, 3, and 11 were used for discriminating graphs with the same decay rate $\lambda = -0.5$ but very different inner-core probabilities. Labels 0, 12, and 13 were used as a case for same λ and similar large inner-core probabilities, and labels 3, 14, and 15 provided an analogous case for similar small inner-core probabilities. Labels 5 and 16 were used for test- ing classification when inner-core probabilities are small and similar with identical high rates of decay. Labels 8 and 18 test the same idea for large inner-core probabilities and a lower rate of decay. Networks generated by these models were used in sections 2.3.3 and 2.5.2.	11
2.7	RF Accuracy Scores for Discriminating between Graphs of $k = \{3, 5, 8\}$ on a Flattened Adjacency Matrix Feature Space. Random and Core- periphery graphs are shown to be the easiest to classify when using random forests with flattened adjacency matrices. Keeping in mind that classifying by random chance is equivalent to rolling a three-sided die ($\frac{1}{3}$ probabil- ity), the accuracy scores in this table suggest that random forests can discriminate between graphs of the same type with different numbers of communities fairly well and better than random chance in all cases	12
2.8	Random Forest Classification Accuracy on Random and Assortative Graphs Sets. See table 2.3 for the parameters corresponding to each label. The first row tests random forests' discriminatory power on a wide range of graph parameters. The second and third rows test binary classification scenarios when graphs have approximately similar density.	13
2.9	Random Forest Classification Accuracy on Disassortative and Ordered Graphs Set. See table 2.3 for the parameters corresponding to each label. The first row tests random forests' discriminatory power on a wide range of graph parameters. For the disassortative graphs, labels 0, 1, and 2 represent probability matrices that have the same diagonal values, but with off-diagonal values that differ by factors of 2. For the ordered graphs, these labels represent probability matrices where the differences between on- and off-diagonal values is approximately 0.2. The last two rows test binary classification scenarios when graphs have approximately similar density.	13
2.10	RF Classification Accuracy Summary for Core-periphery Case, Sub- Experiment 1 . See table 2.4 for a summary of the parameters. The first two rows correspond to classification accuracy when the parameters λ are fairly different. The last two rows correspond to classification scenarios with more similar λ	14
		14

2.11	RF Classification Accuracy Summary for Core-periphery Case, Sub- Experiment 2 . See table 2.5 for a summary of the parameters	15
2.12	RF Classification Accuracy Summary for Core-periphery Case, Sub- Experiment 3. See table 2.6 for a summary of the parameters	15
2.13	RF Accuracy Scores from Discriminating between Graphs of $k = \{3, 5, 8\}$ on a Network Statistics Feature Space. Accuracy scores from classifying on a held-out test set, using the same data and classification set up as described in table 2.7.	20
2.14	RF RF classification scores between graphs of differing <i>M</i> matrices on a network statistics feature space for random and assortative graphs. Accuracy scores from classifying on a held-out test set, using the same data and classification set up as described in table 2.8.	20
2.15	RF classification scores between graphs of differing M matrices on a network statistics feature space for ordered and disassortative graphs. *For ordered graphs, the label set L0, L1, L2 represents graphs whose M matrices have similar differences $M_{ii} - M_{i,i\pm 1}$. In the case of the disassortative graphs, the set L0, L1, L2 corresponds to M matrices with the same on-diagonal values.	21
2.16	RF classification scores for core-periphery graphs, sub-experiment 1, on a network statistics feature space. Reference table 2.4 for a summary of relevant SBM parameters and section 2.3.3 for a description of the experimental set-up.	21
2.17	RF classification scores for core-periphery graphs, sub-experiment 2, on a network statistics feature space. Reference table 2.5 for a summary of relevant SBM parameters and section 2.3.3 for a description of the experimental set-up.	21
2.18	RF classification scores for core-periphery graphs, sub-experiment 2, on a network statistics feature space. Reference table 2.6 for a summary of relevant SBM parameters and section 2.3.3 for a description of the experimental set-up.	21
3.1	Average Densities of 4- and 5-Block MixNet Realizations. As reflected in this table, the models fitted to the macaque data set produce graphs of roughly the same density. In this case, the average densities over all realizations used for this experiment are shown. Intuitively, one may expect any classifier to perform poorly once graphs achieve a certain level of similarity with respect to their densities, particularly if one notices that many of our features in section 2.4.1 are closely related to graph density. However, as shown in Caceres et al. (2016) and later in this section, random forests discriminatory power remains quite strong as long as the underlying edge-probabilities remain relatively distinct.	30

LIST OF FIGURES

1.1	Methodology Pipeline. A schematic of the methods implemented in this research. The top branch is a framework for experiments on synthetic data sets. The lower branch outlines the steps for using random forests to select the best generative model for real data. The arrow connecting "real" to "Random Forest Classifier" means once the classifier is trained on instances from the M models, we input the original data set into the alongifier. The alongifier will then methods the ariginal data to the model it	
	deems most similar.	4
2.1	Table 1 in Caceres et al. (2016). This figure presents the parameters usedfor testing the random forests' ability to discriminate between Erdős-Rényimodels and assortative SBMs.	9
2.2	Summary of Classification Accuracies when Varying Number of Com- munities. Results from tables 2.7 and 2.13.	23
2.3	Summary of Classification Accuracies when Varying SBM Probabil- ity Matrices. Results from tables 2.8, 2.9, 2.14, & 2.15. The lefthand column summarizes random forest classification accuracies on the flat- tened adjacency matrix feature space for experiments using all graph types, excluding core-periphery graphs. The righthand column does the same for the network statistics feature space	24
2.4	Summary of Classification Accuracies when Varying SBM Probabil- ity Matrices (core-periphery graphs). Results from tables providing accuracy scores from both feature spaces for the sub-experiments using core-periphery graph types. Red columns correspond to results from classi- fiers trained on the flattened adjacency matrix feature space, blue columns correspond to those trained on the network statistics feature space	25
3.1	Summary of Mixer Models fitted to the Macaque data set. <i>Top left</i> : ICL vs number of communities per model. The dotted red line indicates that the 4-community MixNet model maximizes ICL and is therefore the best fit to the macaque data set. This model serves as the <i>gold standard</i> with which to compare the our own "best model" chosen by random forests. <i>Top right</i> : Adjacency matrix organized under the best model. <i>Bottom left</i> : Degree distribution. <i>Bottom right</i> : Schematic of probability strength between and within communities under the best model.	27
3.2	Adjacency Matrices of Original Macaque Data with 4- and 5-Block Realizations. Using the methods described in section 3.1, SBMs of 4 and 5 blocks were estimated for the macaque data set. The original adjacency matrix (left) and two realizations of the 4- and 5-block models (top and bottom) are shown.	31
		~ 1

Random Forests as a Model Selection Criterion. Left: Classification				
accuracy versus number of trees for random forests modeled on flattened				
adjacency matrices of the instance graphs and original data set. Right:				
Classification accuracy versus number of trees for random forests modeled				
on network statistics (section 2.4.1) of the instance graphs and original				
data set. For both plots smoothed lines of fit are given, with grey areas				
representing standard error.	33			
	Random Forests as a Model Selection Criterion . <i>Left</i> : Classification accuracy versus number of trees for random forests modeled on flattened adjacency matrices of the instance graphs and original data set. <i>Right</i> : Classification accuracy versus number of trees for random forests modeled on network statistics (section 2.4.1) of the instance graphs and original data set. For both plots smoothed lines of fit are given, with grey areas representing standard error.			

LIST OF ABBREVIATIONS

SBM Stochastic Block Model

- ICL Integrated Completed Likelihood
- RF Random Forests
- MixNet ER (Erdos-Renyi) mixture model for random graphs

CHAPTER 1

Introduction

As sensitive mathematical abstractions of complicated relationships within real-world systems, networks have a wide range of applications in fields such as neuroscience, political science, statistics, and social media (Waugh et al., 2009; Singh et al., 2016; Stam, 2014; Blei et al., 2007; Centola, 2010). Equally extensive are the tools used to further understand networks, from community detection in multilayer networks to network classification with deep learning (Mucha et al., 2010; Yanardag and Vishwanathan, 2015). In this thesis, we combine two popular network science sub-fields: *network* classification and model selection. In network classification, the goal is to use supervised learning *methods* to build models that can differentiate between networks of differing properties. For example, suppose one is given a social networks data set of movie stars in Hollywood, where nodes represent actresses or actors and edges are drawn between them if they appear in the same film. In a supervised *classification* scenario, each network would have a known label, perhaps denoting the genre of the movie. The task is to then take advantage of the networks' properties to build a classifier that can distinguish between the different genres/networks. Model selection in network science focuses on how one can extract probabilistic information about the large-scale structures that may occur within networks in order to define a generative model that can be used to generalize the data or make predictions. If an observed network instance can be mapped to a model, the underlying principles governing the data are easier to identify, however, this is not a trivial task. Our work in this thesis seeks to scale the problem of model selection into a more intuitive framework by converting the issue into a networks classification scenario and using random forests (Breiman, 2001) to select the best model from a set of candidates. We propose that framing model selection in this manner has the combined benefit of 1) taking advantage of the well-studied area of machine learning classification in the context of network science, 2) using popular and simple-to-understand network statistics as

a feature space, and 3) using the built-in feature ranking aspect of random forests to potentially alleviate the "black box" problem that can occur when trying to understand why a particular model has been chosen as a best fit to the data.

We conclude Chapter 1 with a brief overview of some of the literature in network model selection and classification, with more in-depth descriptions provided later as needed, as well as a statement about our motivation and goals. In Chapter 2, we introduce a type of generative model for networks called the Stochastic Block Model (SBM) and describe the discriminatory power of random forests when classifying instances of these models. In Chapter 3, we describe a subtype of the SBM and fit several to a brain connectivity networks data set, using random forests as a model selection criteria and comparing it to an established metric for model selection called Integrated Completed Likelihood (ICL). Our conclusions and future directions are presented in Chapter 4.

1.1 Overview of Network Classification and Modeling

In network classification, the data is assumed to be comprised of networks belonging to the same general space, such as protein-protein interactions or actresses in Hollywood. Within this space there is assumed to exist a finer scale of classification, where within each class networks share more similar properties than with networks from a different class. The goal is to use known instances of each class in order to identify which features best distinguish networks from one another and to use these to build a classifier. Some researchers employ *automated feature selection* and classification methods such as deep learning with kernel functions, while others rely on manual feature selection and extraction, using traditional supervised machine learning techniques such as k-nearest neighbors, support vector machines, or random forests (Yanardag and Vishwanathan, 2015; Canning et al., 2017; Li et al., 2012; Barnett et al., 2016; Caceres et al., 2016). In section 2.4, we describe the pros and cons of using automated versus manual feature extraction and use previous comparative literature to back our choice of manually selecting network statistics as our feature set. Previous research in network classification has shown that random forests performs quite well when compared to other classification algorithms and was used to inform our decision with respect to the use of random forests in this thesis (Barnett et al., 2016; Li et al., 2012). Additionally, research examining the discriminatory power of random forests in classifying instances of SBMs has also shown promising

results, in that random forests can very accurately label instances of SBMs with the correct model for certain graph types, presuming the model parameters are sufficiently different from one another (Caceres et al., 2016).

The sheer number of possible underlying structures that occur in networks has lead to a variety of techniques designed to develop generative models aimed at capturing these properties. Some methods focus on taking advantage of any community structure inherent in the network to devise a metric for community detection (Fortunato, 2010; Newman, 2006; Girvan and Newman, 2002), while others use empirical data to estimate the parameters governing a network's underlying large scale structure (Daudin et al., 2008; Holland et al., 1983; Airoldi et al., 2008). A particularly intriguing model selection criterion is the one proposed by Peixoto (2015). Peixoto notes the difficulties that arise when attempting to compare models that result in diverging descriptions of the same network. To compensate, Peixoto proposes a model selection procedure based on the minimum description length principle. As in this thesis, Peixoto also tests this principle using the stochastic block model and its variants, as well as on a number of empirical network data sets, illustrating the efficiency and scalability of his algorithm. In this thesis, we chose to use the generative model structure proposed by Daudin et al. (2008) and used the model selected by the modified selection criteria defined by Daudin et al. (2008) and originally established by Biernacki et al. (2000) as our gold standard with which to compare random forests' performance. Further information describing Daudin et al. (2008)'s work is presented in section 3.1.2.

1.2 Motivation and Goals

The motivation for this research was inspired by the following question: given a set of *N* generative models fit to a networks data set, can a random forests classifier trained on a fixed number of instances from each model choose the best model when asked to classify the original data set? Research conducted by Caceres *et al.* in *A Model Selection Framework for Graph-based Data* asked a similar question in a narrower context, exploring the discriminatory power of random forests within the parameter space of the Erdős-Rényi and simple stochastic block models (SBM) and using a framework similar to figure 1.1. While thorough in exploring the behavior of random forests with respect to these two simple models, this research did not account for the variety of SBM types that



Figure 1.1: **Methodology Pipeline.** A schematic of the methods implemented in this research. The top branch is a framework for experiments on synthetic data sets. The lower branch outlines the steps for using random forests to select the best generative model for real data. The arrow connecting "real" to "Random Forest Classifier" means once the classifier is trained on instances from the M models, we input the original data set into the classifier. The classifier will then match the original data to the model it deems most similar.

can occur, nor did it explore how to use random forests for choosing the best generative model for a real data set. This honors thesis aims to incorporate these additional facets that, to the knowledge of the authors, have not been examined in the previous research.

Over the course of this research, we implemented the schematic shown in figure 1.1 using two different feature sets, described later in Chapter 2. The overall methodology for the experiments in Chapter 2 is encapsulated by the top "synthetic" branch of the pipeline in figure 1.1. The goal in implementing this branch was to perform a series of experiments designed to provide some intuition about how random forests distinguishes different stochastic block models when trained on a set of realizations from each model. The bottom "real" branch in this figure gives an overview of the methodology used in the work presented in Chapter 3, with the goal of analyzing the effectiveness of framing model selection as a networks classification problem using random forests.

CHAPTER 2

Experiments with Synthetic data sets

To the current knowledge of the authors, the extent of literature exploring the behavior of random forests in the context of network classification and model selection amounts only to the work conducted by Caceres et al. (2016), whose findings are discussed in section 2.2. The key differences between our results and those presented by Caceres et al. (2016) are 1) we explore a much more substantial range of graph types and parameters and 2) we use random forests as a model selection criterion and compare this to other accepted model selection methods (see Chapter 3). Other related research conducted by Canning et al. (2017) and Barnett et al. (2016) that uses random forests for network classification focused more on comparing random forests to other machine learning techniques or simply determining the discriminatory power of network classification by random forests classification in the context of networks.

In this chapter, we implemented two general sets of network classification experiments using random forests with the goal of documenting and understanding the circumstances under which these classifiers perform well or poorly. The first set of experiments focused on classifying graphs of the same *type* (defined in section 2.1) where each type was further categorized by differing numbers of communities. The second set of experiments tested classification accuracy on graphs of the same type and same number of communities, but generated by probability matrices of varying parameter combinations. For each set of experiments, a suite of SBMs using parameters designed to give a full range of possibilities was designed. Synthetic graph data sets were then created by generating 100 instances graphs per SBM. Each graph was assigned a unique, arbitrary label matching it to the parent SBM. These general sets of experiments were implemented twice, the first time using raw edge-weights as the feature set (section 2.3), the second time using a set of network statistics

(section 2.5). The feature sets were split multiple times into training and test sets used to build and test classification accuracy of a random forest model. For each split, the models were trained over a parameter space of $n_{trees} = \{100, 200, 300, 400, 500\}$ using 10-fold cross-validation to select the best model for calculating prediction accuracy.

2.1 Introduction to Stochastic Block Models

Aaron Clauset's Network Analysis and Modeling online lecture notes provided the main template for determining which type of SBM structures to employ in these analyses (Clauset, 2013). Using Clauset's notation, the general stochastic block model is comprised of a $k \times k$ probability matrix M where the entry i, j gives the probability of a vertex in community i connecting to a vertex in community j and k represents the number of communities in the model. SBMs typically also include an $n \times 1$ scalar vector that stores the community membership of a node, where n is the number of nodes in the graph. To generate a graph instance of an SBM, simply loop through each i, j'th pairing and generate an edge with probability M_{ij} . In other words, the probability for each element in a graph adjacency matrix A generated from an SBM with probability matrix M is defined as,

$$P(A_{ij} = 1) \sim \text{Bernoulli}(M_{ij}) \tag{2.1}$$

Clauset's lecture identifies five basic SBM structures which are used as the basis for these experiments, referred to in this paper as *graph types* or *type* (Clauset, 2013).

1. *Random Graphs*: Another name for the Erdős-Rényi graph model, where edge-probabilities are the same for all nodes in all communities.

$$M_{ij} = p \operatorname{constant} \forall i, j \in \{1, \dots, k\}$$

$$(2.2)$$

2. *Assortative Graphs*: The classic SBM, where vertices have strong within-edge probabilities and weaker between-edge probabilities. These tend to produce networks with obvious dense clusters that are sparsely connected. The corresponding probability matrices exhibit strong on-diagonal components and relatively weaker values off the diagonal.

$$M_{ii} > M_{ij} \text{ for } i \neq j$$
 (2.3)

3. *Disassortative Graphs*: The opposite of assortative graphs, these produce a structure where a given node is more likely connect to nodes outside of its community than within. These matrices have strong off-diagonal components and weak on-diagonals.

$$M_{ii} < M_{ij} \text{ for } i \neq j \tag{2.4}$$

4. Ordered Graphs: These are similar to assortative graphs in that vertices of the same community are more likely to be connected, with the additional characteristic of being closely connected to adjacent communities. In terms of matrix structure, these SBMs resemble assortative graphs with strong first-off-diagonal components.

$$M_{ii} \approx M_{i,i-1} \approx M_{i,i+1} \tag{2.5}$$

5. *Core-periphery Graphs*: The core-periphery structure is a subtype of ordered graph, where the probability of an edge decreases exponentially with community index. It may be helpful to think of the probability matrix as containing one large "core" probability which serves as the initial quantity in a system subject to an exponential decay by which the subsequent probabilities are defined. This creates a *nested core structure* where community densities decrease with community index. In general terms, *M* is defined as

$$M_{k \times k} = \begin{cases} P_0 e^{-\lambda(j-1)} & \text{when } j \text{ is on the diagonal} \\ P_0 e^{-\lambda(j)} & \text{when } j \text{ is off the diagonal} \end{cases}$$

where $P_0 = M_{1,1}$, λ is the rate of probability decay, and j = 1, ..., k is a column index. Here we also briefly provide a rough intuition behind the effect of core-periphery parameters λ and P_0 . As $|\lambda| \rightarrow \infty$, the edges located in the outer-cores of the graph begin to drop of exponentially, resulting in single nodes surrounding a small core of nodes connected with probability P_0 . As $|\lambda| \rightarrow 0$, subsequent probabilities M_{ij} approach P_0 , making the network more closely resemble a random structure. Large values of P_0 typically mean larger values in M_{ij} , which translates into more densely connected graphs. In generating dozens of different SBMs with the core-periphery structure, we also observed, at least from a visual point of view, that the effect of λ on the number of edges in the graph appears to be much stronger than that of P_0 .

2.2 Motivation for Synthetic Experiments and Description of SBM Parameters

The goal of the research presented in this chapter is to observe the behavior of random forests in classifying graphs according to their corresponding SBM under a variety of scenarios. To the knowledge of the authors, little has been to done in the way of exploration when it comes to understanding how graphs of different types are distinguished from the perspective of machine learning techniques. Research conducted by Caceres et al. (2016) touches on some of this, exploring the discriminatory power of random forests in the context of classifying Erdős-Rényi and assortative graphs, whose parameters are given in figure 2.1. In particular, Caceres et al. (2016) noticed that classification accuracy decreases as the distance between SBM parameters, $\delta = p_{in} - p_{out}$, shrinks. Chapter 2 expands on this idea, presenting a much larger set of SBM types using a variety of different parameters that are summarized in the tables located in this section. Note that the "Label" column refers to the label provided to random forests for later classification, thus each label corresponds to groups of graphs generated from the same SBM. Each parameter has been chosen with the aim of providing settings where random forests must discriminate between graphs of largely different parameters to graphs of very similar parameters. With the ultimate goal being model selection, we also defined some parameters designed to produce graphs with approximately the same density, as intuition suggests that candidate models for a data set would also produce graphs of similar density. The instance graphs generated from these models were used both in the experiments conducted in sections 2.3 and 2.5. A summary of the total number of SBMs and corresponding instance graphs is provided in table 2.1. Note that the set of SBMs for the core-periphery graph case is particularly extensive due to the wide range of parameter combination possibilities available. As a result, we

Model Type	Parameters
Erdös Dányi	$n = 1000 \ n = 01 \ 015 \ 00$
Ciulos-Kenyi	$n = 1000, p = .01, .015, \dots, .09$
Stochastic block	$n = 1000, p_{in} = .01, .015, \dots, .19, p_{out} = .01, .015, \dots, .09$

Figure 2.1: **Table 1 in** *Caceres et al. (2016)*. This figure presents the parameters used for testing the random forests' ability to discriminate between Erdős-Rényi models and assortative SBMs.

Total SBMs	Total Graphs			
58	5800			

Table 2.1: **Summary of Total Number of SBMs and Instance Graphs**. For each SBM, we generated 100 instance graphs for use in evaluating the discriminatory power over these sets of SBM parameters.

decided to break up the set of core-periphery graphs of varying probability matrices into three general categories. The parameter summaries for each category are presented in tables 2.4, 2.5 and 2.6.

2.3 Experiments Using Flattened Adjacency Matrices as a Feature Space.

2.3.1 Using flattened adjacency matrices as feature sets

For these experiments, random forest models were trained using graph edge-weights, specifically the graphs' flattened adjacency matrices, as the feature set. In other words, suppose synthetic graph G_i is generated from an SBM labeled \mathcal{L}_j , where G_i is undirected and unweighted with adjacency matrix

Graph Type	Parameters (fixed for each $k = 3, 5, 8$)
Random	n = 50, p = 1/k
Assortative	$n = 50, p_{ii} = 0.5, p_{ij} = 0.01 \ j \neq i$
Disassortative	$n = 50, p_{ii} = 0.01, p_{ij} = 0.12 \ j \neq i$
Ordered	$n = 50, p_{ii} = 0.5, p_{i,j\pm 1} = 0.3$
Core-periphery	$n = 50, P_{0,0} = 0.7, \lambda = -0.50$

Table 2.2: Summary of SBM parameters used for classification of graphs of same type, where only the number of communities differ. Networks generated by these models were used in sections 2.3.2 and 2.5.1. n is the number of nodes for each graph, p_{ij} represents the (i, j)th entry in the probability matrices. In the case of the core-periphery structure, λ is the rate of exponential decay as defined earlier. The parameters summarized here represent total of 15 SBMs with three models per graph type, one for each k.

Graph Type	Label 0	Label 1	Label 2	Label 3	Label 4
Random $(n = 50, k = 5)$	p = 0.035	p = 0.05	p = 0.1	p = 0.2	p = 0.5
Assortative $(n = 100, k = 5, p_{i,i} = 0.5)$	$p_{i,j} = 0.01$	$p_{i,j} = 0.02$	$p_{i,j} = 0.05$	$p_{i,j} = 0.10$	$p_{i,j} = 0.25$
Disassortative ($n = 50, k = 5, p_{i,i} = 0.5$)	$p_{i,i} = 0.01$	$p_{i,i} = 0.01$	$p_{i,i} = 0.01$	$p_{i,i} = 0.011$	$p_{i,i} = 0.012$
	$p_{i,j} = 0.12$	$p_{i,j} = 0.24$	$p_{i,j} = 0.06$	$p_{i,j} = 0.058$	$p_{i,j} = 0.28$
Ordered $(n = 75, k = 5)$	$p_{i,i} = 0.5$	$p_{i,i} = 0.3$	$p_{i,i} = 0.7$	$p_{i,i} = 0.31$	$p_{i,i} = 0.68$
	$p_{i,j\pm 1} = 0.2$	$p_{i,j\pm 1} = 0.1$	$p_{i,j\pm 1} = 0.5$	$p_{i,j\pm 1} = 0.085$	$p_{i,j\pm 1} = 0.55$

Table 2.3: Summary of SBM parameters used for classification of graphs of the same type, where each label corresponds to a probability matrix with different parameter values. See tables 2.4, 2.5, and 2.6 for summary of the core-periphery structure parameters. Networks generated by these models were used in sections 2.3.3 and 2.5.2. For all graph types we have fixed the number of communities at 5. In the case of the random graphs, values of p have been chosen to generate graphs with a range of densities. Smaller values of p indicate low edge probability, resulting in sparser networks. Conversely, as p increases, the network density grows as well. For assortative graphs, we have fixed inter-community probability as 0.5 and vary the between-community edge probabilities (p_{ij} where $j \neq i$). As p_{ij} increases, the assortative structure more closely resembles that of a random graph. The disassortative parameters mimic the assortative case, in that we only vary p_{ij} . Values of p_{ij} are selected such that nodes in graphs from differing labels are twice as likely or half as likely to connect to nodes in other communities. Other parameters for the disassortative graphs are selected as dense and sparse counterparts. The parameters for the ordered graphs are selected such that two labels have the same value for $p_{ii} - p_{i,i\pm 1}$ and other labels are slightly sparser or denser versions.

Label	P_0	λ
Label 0	0.7	-0.5
Label 1	0.7	-0.2
Label 2	0.7	-0.7
Label 3	0.55	-0.5
Label 4	0.55	-0.2
Label 5	0.55	-0.7
Label 6	0.7	-0.45
Label 7	0.55	-0.45

Table 2.4: **Parameter Summary for Core-periphery Case, Sub-Experiment 1**. (Reference *item* 5 in section 2.1 for notation.) Networks generated by these models were used in sections 2.3.3 and 2.5.2. Labels 0, 1, and 2 correspond to networks with strong inner-core edge probability (P_0) with varying rates of decay (λ). Labels 3, 4, and 5 test the same idea with a weaker inner-core edge probability. Labels 6 and 7 were used to test classification problems where networks have the same inner-core edge probabilities and similar rates of decay. They were tested with labels 0 and 3 respectively.

Label	P_0	λ
Label 8	0.7	-0.3
Label 9	0.72	-0.32
Label 10	0.58	-0.52

Table 2.5: **Parameter Summary for Core-periphery Case, Sub-Experiment 2**. Labels 8 and 9 were used for testing discriminatory power in the case of similar cores and decay rates when the networks are dense. Labels 10 and 3 were used for the same purpose except for sparse networks. Networks generated by these models were used in sections 2.3.3 and 2.5.2.

Label	P_0	λ
Label 11	0.85	-0.5
Label 12	0.74	-0.5
Label 13	0.78	-0.5
Label 14	0.59	-0.5
Label 15	0.63	-0.5
Label 16	0.57	-0.7
Label 17	0.73	-0.3

Table 2.6: **Parameter Summary for Core-periphery Case, Sub-Experiment 3**. Labels 0, 3, and 11 were used for discriminating graphs with the same decay rate $\lambda = -0.5$ but very different innercore probabilities. Labels 0, 12, and 13 were used as a case for same λ and similar large inner-core probabilities, and labels 3, 14, and 15 provided an analogous case for similar small inner-core probabilities. Labels 5 and 16 were used for testing classification when inner-core probabilities are small and similar with identical high rates of decay. Labels 8 and 18 test the same idea for large inner-core probabilities and a lower rate of decay. Networks generated by these models were used in sections 2.3.3 and 2.5.2.

 $A_i \in \mathbb{R}^{n \times n}$. The corresponding feature vector for \mathcal{G}_i , the so-called "flattened adjacency matrix", is a row vector of zero's and one's.

$$\operatorname{vec}(A_i)^T = \mathcal{F}_i \in \mathbb{R}^{1 \times (n \times n)}$$
(2.6)

2.3.2 Classification of graphs of same type, where only the number of communities differ.

For each graph type in section 2.1, three sets of SBMs were defined. The only parameter that varied was the number of communities, k, where k = 3, 5, 8. Table 2.2 is a summary of the graph types and corresponding parameters that were fixed as k varied.

For each SBM, 100 instance graphs were generated and their flattened adjacency matrices \mathcal{F}_i extracted. An instance graph was labeled by the number of communities of the corresponding parent SBM. Random forest classifiers were trained on 67% of the data and accuracy scores from classifying the held-out 33% were used to estimate the models' test accuracies. Generally speaking, random forests seem able to discriminate fairly accurately between SBM instances of the same type with differing numbers of communities. Visually speaking, the structural differences between these graph classes were easy to spot. In many cases, strong classification accuracies seemed to follow the cases where graph instances of different communities exhibited very different structures. The

Graph Type	Accuracy Score
Random	87.85 ± 10.66
Assortative	61.89 ± 16.11
Disassortative	38.11 ± 19.36
Ordered	81.0 ± 15.86
Core-periphery	83.89 ± 12.76

Table 2.7: **RF Accuracy Scores for Discriminating between Graphs of** $k = \{3, 5, 8\}$ **on a Flattened Adjacency Matrix Feature Space.** Random and Core-periphery graphs are shown to be the easiest to classify when using random forests with flattened adjacency matrices. Keeping in mind that classifying by random chance is equivalent to rolling a three-sided die ($\frac{1}{3}$ probability), the accuracy scores in this table suggest that random forests can discriminate between graphs of the same type with different numbers of communities fairly well and better than random chance in all cases.

one exception, seen in the low accuracy score of the disassortative experiment, corresponded to a case where it was very difficult to visually distinguish the graph types. A likely contributing factor to the lower classification score is that the parameter values of the corresponding SBMs of these graphs varied perhaps the least out all the experiments. Both the results in Caceres et al. (2016) and subsequent experiments in section 2.5 also suggest that the discriminatory power of random forests decreases as SBM parameters become more similar.

2.3.3 Classify graphs of the same type, where each label corresponds to a probability matrix with different parameter values.

The possibilities for combinations of different SBM parameters within a graph type resulted in a slightly more involved series of experiments than those described above. For each graph type, a set of SBMs were defined in such a way to give a wide range of graph parameters. As in section 2.3.2, 100 instance graphs were generated for each of the SBMs used in an experiment and random forest classifiers were trained on 67% of the data and tested on the remaining 33%. Graphs were assigned a unique label corresponding to their parent SBM. The following discussion summarizes each experiment (corresponding to graph type).

Summarizing experiments using Random, Assortative, Dissassortative, and Ordered Graphs: The results for these experiments are given in tables 2.8 and 2.9. For both tables 2.8 and 2.9, accuracy was the lowest when random forests were tasked to classify graphs from wide range of parameters. The subsequent rows in table 2.8 show as the distance between the parameters increases, classification

Label Set	Random	Assortative
L0, L1, L2, L3, L4	60.22 ± 11.07	43.00 ± 09.71
L0, L1 (sparse case)	67.46 ± 18.97	49.37 ± 15.78
L2, L3 (dense case)	95.08 ± 8.71	90.87 ± 10.52

Table 2.8: **Random Forest Classification Accuracy on Random and Assortative Graphs Sets**. See table 2.3 for the parameters corresponding to each label. The first row tests random forests' discriminatory power on a wide range of graph parameters. The second and third rows test binary classification scenarios when graphs have approximately similar density.

Label Set	Disassortative	Ordered
L0, L1, L2, L3, L4	46.56 ± 11.22	48.60 ± 11.24
L0, L1, L2	66.70 ± 16.13	91.96 ± 8.73
L2, L3 (sparse case)	52.30 ± 18.13	53.57 ± 22.06
L1, L4 (dense case)	53.73 ± 21.58	55.40 ± 22.07

Table 2.9: **Random Forest Classification Accuracy on Disassortative and Ordered Graphs Set.** See table 2.3 for the parameters corresponding to each label. The first row tests random forests' discriminatory power on a wide range of graph parameters. For the disassortative graphs, labels 0, 1, and 2 represent probability matrices that have the same diagonal values, but with off-diagonal values that differ by factors of 2. For the ordered graphs, these labels represent probability matrices where the differences between on- and off-diagonal values is approximately 0.2. The last two rows test binary classification scenarios when graphs have approximately similar density.

accuracy also increases. Differences between the parameters for L0 and L1 were not as large as differences between parameters for L2 and L3 due to the relatively smaller probability values for the sparser case. As a result, we see the strongest classification scores in the final row of table 2.8. A similar case can be made for table 2.9, except notice that the largest differences in parameters occurred for the L0, L1, L2 label set, rather than the sparse and dense label sets.

In general, the results in tables 2.8 and 2.9 tell us that when parameters of SBMs are selected to produce instance graphs of similar parameters, even if graph density is similar, random forests' classification accuracy decreases. However, for sufficient differences in the parameters, even using a feature set as trivial as flattened adjacency matrices allows for strong classification accuracy. For example, consider the classification accuracies for the ordered graph types in table 2.9. When tasked to classify sets of graphs with similar parameters (L2 & L3 and L1 & L4), the random forest models performed barely better than a coin toss. However, when classifying a set of graphs with a wide enough range of parameters (label set L0, L1, L2), classification accuracy jumped to nearly 92 percent.

Label Set	Accuracy
L0, L1, L2 (Same strong core density, wide range of decay rates)	83.22 ± 11.36
L3, L4, L5 (Same weak core density, wide range of decay rates)	72.26 ± 14.68
L0, L6 (Same strong core density, similar decay rates)	49.52 ± 20.15
L3, L7 (Same weak core density, similar decay rates:)	59.92 ± 21.17

Table 2.10: **RF Classification Accuracy Summary for Core-periphery Case, Sub-Experiment 1**. See table 2.4 for a summary of the parameters. The first two rows correspond to classification accuracy when the parameters λ are fairly different. The last two rows correspond to classification scenarios with more similar λ .

Experiments with Core-Periphery graphs: The set of SBMs for the core-periphery graph case is particularly extensive due to the wide range of parameter combination possibilities available. Recall that this particular core-periphery structure is defined by exponentially decreasing probabilities indexed by community. As magnitude of the rate of decay approaches infinity, all that is left is the "core" community with edge-probabilities of nodes in the outer-cores approaching zero and overall density decreasing. As the rate of decay magnitude moves in the opposite direction, approaching zero, overall graph density increases and the core-periphery structure acts more like a random graph, with outer-core edge-probabilities staying very close to the inner-core P_0 . In other words, an increased rate of decay results in a decrease in graph density and a decrease in decay rates results in an increase of graph density. Conversely, larger P_0 values give denser graphs, whereas smaller P_0 results in weaker out-core probabilities. In order to take into account the various interactions at play, we devised the three following sub-experiments.

Core-periphery case, sub-experiment 1: Here we defined 8 SBMs designed to provide classification scenarios where random forests discriminated between core-periphery graphs with the same P_0 and differing rates of decay. From the last two rows of table 2.10, one can observe how the discriminatory power of random forests decreases when it is tasked with classifying graphs of similar λ rates. Otherwise, classification accuracy was fairly strong for a wider range of parameter differences.

Core-periphery case, sub-experiment 2: Here we compared 2 sets of graphs defined by similar P_0 and λ values. As expected, classification accuracies were quite low, with both worse than a random coin toss (table 2.11).

Core-periphery case, sub-experiment 3: This sub-experiment compared core-periphery structures of same rates of decay λ and differing core-probabilities P_0 . Overall, classification accuracies were

Label Set	Accuracy
L8, L9 (dense case)	46.51 ± 22.48
L3, L10 (sparse case)	47.06 ± 20.02

Table 2.11: **RF Classification Accuracy Summary for Core-periphery Case, Sub-Experiment 2**. See table 2.5 for a summary of the parameters.

Label Set	Accuracy
L0, L3, L11 ($\lambda = -0.5$, different P_0)	48.89 ± 15.92
L0, L12, L13 ($\lambda = -0.5$, similar large P_0)	31.74 ± 15.53
L3, L14, L15 ($\lambda = -0.5$, similar small P_0)	33.67 ± 15.57
L5, L16 ($\lambda = -0.7$, sparse case)	43.65 ± 17.48
L8, L18 ($\lambda = -0.3$, dense case)	55.08 ± 12.86

Table 2.12: **RF Classification Accuracy Summary for Core-periphery Case, Sub-Experiment 3.** See table 2.6 for a summary of the parameters.

much lower for these scenarios, most likely due to the fact that the rate of decay has a stronger influence on the graph structure than P_0 , thus graphs are more similar to one another when their respective λ 's are also similar, regardless of their P_0 values.

2.3.4 Summary of results on the flattened adjacency feature space

Generally speaking, when classifying graph instances from models of sufficiently different parameters, random forests were often fairly adept at distinguishing graphs from most SBM types. Clearly, however, accuracy scores varied depending on which graph type was the focus of the experiment. For example, classification scores for ordered graphs in both general experiments varying number of communities and varying probability matrix values were usually quite strong, whereas disassortative graphs seemed to produce lesser accuracy scores in all cases. The relationship between the parameter differences and random forest classification accuracy was also noted by Caceres et al. (2016) for Erdős-Rényi models (random graphs) and the assortative SBM structure.

2.4 Using Network Statistics

In section 2.3, we employed perhaps the most trivial method for converting graphs into a numeric feature space for classification by mapping each graph \mathcal{G}_i to vector \mathcal{F}_i , where $\mathcal{F}_i = \text{vec}(A_i)^T$. In other words, section 2.3 explored graph classification using one-dimensional vectors of binary variables, where each variable denoted the existence of an edge between all given node pairings.

This method worked adequately for graphs from sufficiently different parameter spaces, however, in the cases where graph parameters became more similar, using the flattened adjacency matrix feature space often failed to provided models with strong discriminatory power. Perhaps more importantly, the representation of graphs by their flattened adjacency matrices lacks interpretability.

The problem of feature representation in the context of graph classification has recently attracted many researchers and has resulted in several alternative methods for mapping graphs to an adequate feature space (Li et al., 2012; Barnett et al., 2016; Caceres et al., 2016; Canning et al., 2017; Yanardag and Vishwanathan, 2015) (see also "Related Work" in Li et al., 2012). Many have explored the use of a graph's topological properties, also known as "network statistics" in the field of network science, as an effective means of representing graph data sets for classification (Newman, 2010; Li et al., 2012; Barnett et al., 2016; Caceres et al., 2016; Canning et al., 2017). Another alternative to defining a feature space with network statistics is to employ kernel methods. As defined in Shawe-Taylor and Cristianini (2004), the appeal of using kernel methods is that the so-called kernel function can bypass feature-vector representation by calculating the inner products between the projections of data pairings into the feature space without computing their actual coordinates in said feature space. In other words, a kernel function is a direct inner product of the input features that avoids explicitly mapping to a feature space. To paraphrase the notation defined in Shawe-Taylor and Cristianini (2004), a kernel function is

$$\kappa(\mathcal{G}_i, \mathcal{G}_j) = \langle \phi(\mathcal{G}_i), \phi(\mathcal{G}_j) \rangle \tag{2.7}$$

where ϕ is a mapping to feature space $F \subseteq \mathbb{R}^N$

$$\phi: \mathcal{G}_i \to \phi(\mathcal{G}_i) \in F \tag{2.8}$$

This method appealingly makes accessible feature spaces that are exponentially or even infinitely large (Shawe-Taylor and Cristianini, 2004). However, some comparative analyses suggest that not only does the use of a feature space defined by network statistics yield comparable to more accurate graph classification results for a variety of discriminatory algorithms, calculating such a feature space also takes substantially less time than other kernel methods (Barnett et al., 2016; Li et al., 2012). Because many network statistics, such as average degree, are intuitively easy to connect to the "real

world" properties of a graph, classification based off these features has the additional benefit of clear interpretability, particularly if the classifier automatically identifies the features most relevant to classification (such as random forests). As a result, we have decided to explore model selection using graph classification on a feature space defined by a variety of network statistics described in section 2.4.1.

2.4.1 Description of network statistics

The following list of thirty-seven network statistics (including the minimums, maximums, averages, and standard deviations of certain measurements) has been compiled using the variety of relevant literatures listed in the introduction of this section. Generally speaking, items 3 - 10, 16 - 18 comprise of measurements that encapsulate notions of node degree and "connectedness" of a graph, items 11 - 15, 24, and 25 employ information about shortest paths, and items 19 - 23 use information encoded in the graphs' adjacency matrices. For the measurements using eccentricity (items 11 - 13) and for item 14, we calculated averages weighted by number of nodes per component if the graphs were disconnected (Li et al., 2012).

We would like to credit the follow recent works for inspiring the use of these network statistics: Li et al. (2012) for items 7 - 23 and Caceres et al. (2016) for items 3 - 6, 24, and 25. All statistics were computed using the Python module NetworkX (Hagberg et al., 2008).

- 1. Number of Nodes: Total number of nodes in the graph.
- 2. Number of Edges: Total edge count.
- 3. **Number of Triangles**: A triangle is defined as a complete graph consisting of three nodes and three edges. The total number of triangles is defined with respect to the entire graph.
- 4. Maximum Triangles: The maximum number of triangles for a single node in the graph.
- 5. Average Triangles: The average number of triangles to which a single node belongs.
- 6. Standard Deviation of Triangles: The corresponding standard deviation.

- Global Clustering Coefficient: The number of closed triplets (3 × total number of triangles) divided by the total number of connected three-node subgraphs. See Fig 1. in Li et al. (2012). This statistic provides an overall measure of clustering for the entire graph.
- 8. Local Clustering Coefficient: As implied by the name, the local clustering coefficient is defined same as above, this time with respect to a single node. Essentially, this measure quantifies the amount of "connectedness" obtained by a given node and its' neighbors. We compute the average, standard deviation, minimum, and maximum local clustering coefficients for each graph.
- 9. Average Degree: The average degree over all nodes in a graph, where "degree" refers to the number of edges directly adjacent to a given node.
- 10. **Degree Assortivity Coefficient**: A measurement of a node's "preference" for attaching to other nodes with similar degree. In this case, the standard Pearson correlation coefficient is used (see equation 21 in Newman (2003)). Negative values imply that a given high degree node will tend to connect with nodes of lower degree and vice versa. Positive values indicate that high degree nodes tend to connect with other high degree nodes and low degree nodes more frequently connect with other low degree nodes.
- 11. Average Eccentricity: The maximum distance from a node v to all other nodes in the graph.
- 12. Radius: Minimum eccentricity.
- 13. Diameter: Maximum eccentricity.
- 14. **Percentage of Central Points**: Ratio of nodes with minimum eccentricity over total nodes in the graph.
- 15. Closeness Centrality: For a given node v, closeness centrality is the reciprocal of the sum of the lengths of the shortest paths. Larger centrality measures correspond to more "central" nodes. The NetworkX module normalizes these scores by multiplying by total nodes minus one.
- 16. **Giant Connected Ratio**: Ratio of number of nodes in the largest connected component to total nodes in the graph.

- 17. Percentage of Isolated Nodes: Nodes with degree equal to zero expressed as a percentage.
- 18. Percentage of Endpoints: Nodes with degree equal to one expressed as a percentage.
- 19. Spectral Radius: Eigenvalue of the largest magnitude in the adjacency matrix.
- 20. Second Largest Eigenvalue: Eigenvalue of the second largest magnitude in the adjacency matrix.
- 21. **Trace of Adjacency Matrix**: The trace of the adjacency matrix, also known as the sum of the eigenvalues.
- 22. Energy: Squared sum of eigenvalues.
- 23. **Number of Distinct Eigenvalues**: Quantifies the number of distinct eigenvalues of the adjacency matrix. In the undirected case, this should correspond to the total number of nodes.
- 24. **Shortest path**: The length of the path is always 1 less than the number of nodes involved in the path. The shortest paths involving all pairs of nodes are summarized using minimum, maximum, average, and standard deviation.
- 25. Betweenness Centrality: For a given node v, betweenness centrality is the ratio of the number of shortest paths going through v divided by all other shortest paths not including v.

2.5 Experiments Using Network Statistics as a Feature Space

The subsequent parts of section 2.5, subsections 2.5.1 and 2.5.2, closely mirror the previous sections describing random forest classification accuracy using the graphs' flattened adjacency vectors, \mathcal{F}_i . In fact, these sections repeat the classification experiments described in sections 2.3.2 and 2.3.3 except this time using network statistics as the feature space.

2.5.1 Classification of graphs of same type, where only the number of communities differ.

As seen in table 2.13, the discriminatory power of network statistics-based random forest classifier is very strong when working with synthetic graphs. In particular, random forests seemed to

Graph Type	Accuracy Score
Random Graphs	100.00 ± 0.00
Assortative Graphs	99.33 ± 0.95
Disassortative Graphs	59.26 ± 1.26
Ordered Graphs	100.00 ± 0.00
Core-Periphery Graphs	98.65 ± 0.95

Table 2.13: **RF Accuracy Scores from Discriminating between Graphs of** $k = \{3, 5, 8\}$ **on a Network Statistics Feature Space.** Accuracy scores from classifying on a held-out test set, using the same data and classification set up as described in table 2.7.

Label Set	Random Graphs	Assortative Graphs
L0, L1,, L5 (spectrum of edge-probabilities)	96.36 ± 1.31	99.60 ± 0.29
L0&L1 (sparse comparison)	86.36 ± 4.46	100.00 ± 0.00
L2&L3 (dense comparison)	100.00 ± 0.00	100.00 ± 0.00

Table 2.14: **RF RF classification scores between graphs of differing** M **matrices on a network statistics feature space for random and assortative graphs.** Accuracy scores from classifying on a held-out test set, using the same data and classification set up as described in table 2.8.

perform exceptionally well on graph types generated by probability matrices with relatively strong diagonal components or when edge-probabilities are constant throughout (i.e., in the case of random graphs). Note the highest scores correspond to either graphs exhibiting strong community structures (assortative, ordered, and core-periphery) or graphs whose edge-probability is directly dependent on the number of communities (random graphs with $p = p_{in} = p_{out} = \frac{1}{k}$). The lowest score by far corresponds to the disassortative case, which by definition generates communities that have small p_{in} . Comparing results from using flattened adjacency matrices as a feature space (table 2.7) shows that using network statistics yields a minimum 15% increase in accuracy for these experiments. In both cases, random graphs prove to be the easiest to classify, dominating in terms of accuracy in table 2.7 and maintaining the lead in table 2.13. Additionally, graphs with a disassortative structure proved hardest to classify in both cases. However, the general increase in accuracy for assortative, ordered, and core-periphery graphs does not precisely mirror this trend, with assortative graphs moving from fourth to second in terms of accuracy and core-periphery graphs moving from second to third.

Label Set	Ordered Graphs	Disassortative Graphs
L0, L1,, L4 (spectrum of edge-probabilities)	68.89 ± 1.25	73.13 ± 0.76
L0, L1, L2*	100.00 ± 0.00	100.0 ± 0.0
L1, L3 (sparse comparison)	63.64 ± 6.19	51.01 ± 3.11
L2, L4 (dense comparison)	69.19 ± 2.58	86.87 ± 1.89

Table 2.15: **RF classification scores between graphs of differing** M matrices on a network statistics feature space for ordered and disassortative graphs. *For ordered graphs, the label set L0, L1, L2 represents graphs whose M matrices have similar differences $M_{ii} - M_{i,i\pm 1}$. In the case of the disassortative graphs, the set L0, L1, L2 corresponds to M matrices with the same on-diagonal values.

Label Set	Accuracy
L0, L1, L2 (Same strong core density, wide range of decay rates)	99.33 ± 0.48
L3, L4, L5 (Same weak core density, wide range of decay rates)	96.63 ± 1.26
L0, L6 (Same strong core density, similar decay rates)	72.22 ± 4.34
L3, L7 (Same weak core density, similar decay rates:)	72.22 ± 1.89

Table 2.16: **RF classification scores for core-periphery graphs, sub-experiment 1, on a network statistics feature space.** Reference table 2.4 for a summary of relevant SBM parameters and section 2.3.3 for a description of the experimental set-up.

Label Set	Accuracy
L8, L9: same-ish core, same-ish decay rate, dense case	57.58 ± 1.24
L3, L10: same-ish core, same-ish decay rate, sparse case	47.98 ± 4.34

Table 2.17: **RF classification scores for core-periphery graphs, sub-experiment 2, on a network statistics feature space.** Reference table 2.5 for a summary of relevant SBM parameters and section 2.3.3 for a description of the experimental set-up.

Label Set	Accuracies
L0, L3, L11: Same decay rate (-0.5), very different core densities	72.39 ± 0.48
L0, L12, L13: Same decay rate (-0.5), similar strong core density	31.31 ± 6.23
L3, L14, L15: Same decay rate (-0.5), similar weak core density	48.48 ± 2.47
L5, L16: Same decay rate (-0.7), sparse case	49.49 ± 3.11
L8, L18: Same decay rate (-0.3), dense case	61.11 ± 5.85

Table 2.18: **RF classification scores for core-periphery graphs, sub-experiment 2, on a network statistics feature space.** Reference table 2.6 for a summary of relevant SBM parameters and section 2.3.3 for a description of the experimental set-up.

2.5.2 Classification graphs of the same type, where each label corresponds to a probability matrix with different parameter values.

As in subsection 2.5.1, in all cases one can observe substantial gains in classification accuracy on the held-out test sets when compared to classification on a flattened adjacency matrix feature space. As also observed in subsection 2.5.1, some within-experiment accuracy scores maintained their relative positions while others did not. The scenario where an accuracy remains the highest when compared to other experiments is most clearly apparent for the experiments in using ordered graphs (tables 2.9 and 2.15). In other cases, the pattern does not quite hold. Consider tables 2.14 and 2.8. When using flattened adjacency matrices as a feature space, random forests struggled the most when tasked with classifying graphs displaying a spectrum of edge-probabilities. However, when using network statistics, random forests yielded lower classification accuracy scores for the experiment comparing relatively sparse graphs than when discriminating on a spectrum of edge-probabilities.

Similar comparisons can be made for all the relevant tables, with the most relevant conclusion being that, for all cases, the use of network statistics as a feature space results in large gains with respect to classification accuracy. In many cases, under network statistics, random forests achieved perfect discriminatory power. Interestingly, whether the relative changes in accuracy across subexperiments mirrors those displaced in the flattened adjacency cases seems to depend on graph type and the set of graphs random forests is tasked with classifying.

2.6 Summary

The motivation for implementing the experiments presented in Chapter 2 stemmed primarily from a desire to document and understand the general behavior of random forests when trained on different feature spaces under a variety of classification scenarios induced by different combinations of stochastic block model types and parameter combinations. The workflow for these experiments is summarized in the "synthetic" branch of figure 1.1. Our results are summarized in figures 2.2, 2.3, and 2.4. Unsurprisingly, for all but a few cases in the core-periphery experiments (figure 2.4), the use of network statistics as a feature space results in classifiers of discriminatory power much greater than those trained on the flattened adjacency matrix feature space.



Figure 2.2: Summary of Classification Accuracies when Varying Number of Communities. Results from tables 2.7 and 2.13.

Recall that sets of experiments in sections 2.3.2 and 2.5.1 examined random forests' discriminatory power when classifying between models of the same type (i.e. random, assortative, disassortative, ordered, and core-periphery) with the same numbers of nodes and general parameters in the probability matrix. The only differences between these models was the number of communities. As seen in figure 2.2 not only did the use of network statistics result in substantial gains for all graph types, the relative accuracies between graph types was also approximately the same. In other words, classifiers on the flattened adjacency matrix space that performed better for a given graph type versus a different graph type still performed better for the given graph type when trained on the network statistics space.

The second set of experiments (sections 2.3.3 and 2.5.2) examined the discriminatory power of random forests when graph types are the same with same numbers of communities and nodes, but with different underlying edge probabilities. As noted before, when using network statistics as a feature set, the discriminatory power of random forests substantially increases. In many cases random forests achieved perfect classification accuracy scores. Interestingly, the relative change in accuracy when using network statistics as opposed to flattened adjacency matrices as a feature space does not necessarily remain the same as it did for figure 2.2. For example, consider the results



Figure 2.3: **Summary of Classification Accuracies when Varying SBM Probability Matrices.** Results from tables 2.8, 2.9, 2.14, & 2.15. The lefthand column summarizes random forest classification accuracies on the flattened adjacency matrix feature space for experiments using all graph types, excluding core-periphery graphs. The righthand column does the same for the network statistics feature space.

for assortative and random graphs shown in the top row of figure 2.3. When trained on flattened adjacency matrices, random forests had better overall accuracy in discriminating different instances of random graphs than when classifying different assortative graphs. However, when using network statistics as a feature space, higher classification rates occur for the assortative graphs. This trend reversal also occurs when comparing disassortative and ordered graphs (second row of figure 2.3).

While the use of network statistics as a feature space lead to substantial gains in discriminatory power, it should be noted that not all gains were equal across graph types and classification scenarios. In general, it appears that the discriminatory power of random forests does depend on graph type and, as already noted by Caceres et al. (2016), the difference between the underlying graph parameters. From these results, we now have a working idea of under which circumstances random forests can best classify graphs generated by SBMs and can explore random forests' capability as a model selection criterion according to the second workflow outlined in figure 1.1.



Figure 2.4: **Summary of Classification Accuracies when Varying SBM Probability Matrices** (**core-periphery graphs**). Results from tables providing accuracy scores from both feature spaces for the sub-experiments using core-periphery graph types. Red columns correspond to results from classifiers trained on the flattened adjacency matrix feature space, blue columns correspond to those trained on the network statistics feature space.

CHAPTER 3

Model Selection Using Random Forests

This chapter explores the question posed at the beginning of this thesis: given N candidate stochastic block models for a particular networks data set, is it possible for a random forests classifier trained on a set of instance graphs generated from these N SBMs to select the best fit to a real-world network? Our goal is to determine whether this well-known and relatively easy to implement machine-learning classification technique can serve as a comparable method for fitting generative models to a given network's physical and probabilistic structure. We have divided this chapter into two sections. The first provides an overview of the tools and techniques used to determine the N candidate models. This section also describes the real data set used for our experiments. The final section presents our findings in comparing the model selected by random forest to the "gold standard" model selected using the criteria described below.

3.1 Overview of the *R* package 'mixer'and data set

The R package mixer provides routines for unsupervised clustering of networks using a generalized version of the Erdős-Rényi model, the Erdős-Rényi mixture for random graphs (MixNet). Mixer contains a wrapper-function for estimation of the MixNet parameters and automatically applies the Integrated Classification Likelihood criterion to select the best "mixer" model. Additionally, mixer contains the data set *macaque*, a networks representation of a macaque brain containing 47 brain cortical regions (nodes) and 505 inter-regional pathways (edges). According to the package documentation, the mixer model accurately identifies regions within this data set that serve as "hubs" for information flow within the macaque brain (LAPACK et al., 2015). With respect to the terminology used in this thesis, "hubs" for our purposes are communities. In particular, the methods employed by mixer defines communities using nodes that are heavily connected to one another,



Figure 3.1: **Summary of Mixer Models fitted to the Macaque data set.** *Top left*: ICL vs number of communities per model. The dotted red line indicates that the 4-community MixNet model maximizes ICL and is therefore the best fit to the macaque data set. This model serves as the *gold standard* with which to compare the our own "best model" chosen by random forests. *Top right*: Adjacency matrix organized under the best model. *Bottom left*: Degree distribution. *Bottom right*: Schematic of probability strength between and within communities under the best model.

resulting in a pattern strongly reminiscent of the assortative graph structure. Using mixer, the process of fitting several MixNets of differing numbers of communities to the macaque data set is trivialized to a few lines of code and has the additional benefit of allowing us to choose in advance a gold standard with which to compare our random forests results. As shown in figure 3.1, we shall assume that the best model for our data set has 4 communities.

3.1.1 Brief overview of MixNet models

This section provides a brief summary of previous work that derived the tools and methods used to define our set of N models for the macaque data set. A detailed description for the derivation, properties, parameter estimation techniques, and model selection criteria of the Erdős-Rényi mixture

for random graphs can be found in Daudin et al. (2008) and information regarding ICL is presented in Biernacki et al. (2000).

In defining MixNet, Daudin et al. (2008) assumes the mixture model framework for defining the underlying probabilistic structure of a given network. Note that the SBMs presented in section 2.1 are all mixture models and thus are a reflection of the same framework about to be described. Paraphrasing the notation in Daudin et al. (2008), mixture models assume nodes are grouped into K communities with prior probability α_k . Let $\{Z_{ik}\}$ be an indicator variable for the community of node i, then the prior probabilities of node-community membership can be expressed as

$$\alpha_k = Pr\{Z_{ik} = 1\}, \text{ with } \sum_k Z_{ik} = 1, \sum_k \alpha_k = 1$$
 (3.1)

Recall that setting edge-probabilities as $P(A_{ij} = 1) \sim \text{Bernoulli}(p)$ produces the so-called Erdős-Rényi random graph model, G(n, p). Rather than assuming that edges are independent with Bernoulli(p) distributions, Daudin et al. (2008) requires the definition of inter-community probabilities π_{kl} , or the probability that a node in community k connects to a different node in community l. As in the cases outlined in section 2.1, graphs are assumed to be undirected, which means that $\pi_{kl} = \pi_{lk}$. Finally, edges in these graphs are assumed to be conditionally independent of the communities involved and no self-loops are allowed,

$$\begin{cases} A_{ij} | \{Z_{ik} = 1, Z_{jl} = 1\} \sim \text{Bernoulli}(\pi_{kl}) \\ A_{ii} = 0 \end{cases}$$

$$(3.2)$$

The connectivity matrix $\boldsymbol{\pi} = (\pi_{kl})$ in Daudin et al. (2008) is the same as our own probability matrix M defined in section 2.1.

3.1.2 MixNet model estimation

The paper for MixNet models by Daudin et al. (2008) provides extensive proofs concerning parameter estimation. As these are not directly related to the main focus of this thesis, we will let the reader defer to them as needed and limit ourselves to presenting their propositions. The following formulas and corresponding algorithms are implemented in the R package 'mixer'.

To begin, Daudin et al. (2008) defines the log-likelihood of a network defined under the MixNet model as

$$\log \mathcal{L}(\mathcal{X}, \mathcal{Z}) = \sum_{i} \sum_{k} Z_{ik} \log \alpha_k + \frac{1}{2} \sum_{i \neq j} \sum_{k,l} Z_{ik} Z_{jk} \log \operatorname{Bernoulli}(X_{ij}; \pi_{kl})$$
(3.3)

where $\mathcal{X} = \{X_{ij}\}$ is the set of all edges and $\mathcal{Z} = \{Z_{ik}\}$ is the set of all indicator variables defined previously. Note that Bernoulli $(X; \pi) = \pi^X (1-\pi)^{1-X}$. As discussed in the literature, the likelihood $\mathcal{L}(\mathcal{X})$ cannot be simplified into a more tractable form for computation. Instead, Daudin et al. (2008) proposes a variational approach that attempts to optimize the lower-bound of the likelihood function and an iterative algorithm designed to estimate the prior probabilities α and the class-connectivity matrix π while maximizing this lower bound. This algorithm assumes a fixed number of communities K when updating the parameters α and π . To choose the best model given different K, Daudin et al. (2008) uses a modified Integrated Classification Likelihood selection criterion developed by Biernacki et al. (2000). Given a model m_K of K communities, this model selection criterion is defined as

$$\operatorname{ICL}(m_K) = \max_{\boldsymbol{\theta}} \log \mathcal{L}(\mathcal{X}, \widetilde{\mathcal{Z}} | \boldsymbol{\theta}, m_K) - \frac{1}{2} \times \frac{K(K+1)}{2} \log \frac{n(n-1)}{2} - \frac{K-1}{2} \log n \quad (3.4)$$

where $\theta = (\alpha, \pi)$ is the entire set of mixture parameters, \widetilde{Z} are the predictions of Z, and n is the total number of nodes in the model (Daudin et al., 2008). The R mixer() function implements these equations on a given adjacency matrix for a pre-defined range of K. For each of the K models, equation 3.4 is computed. The model with the largest ICL value is selected as the best fit to the data (see panel 1 in figure 3.1).

3.2 Model Selection using Random Forests

Using the tools described in section 3.1, we are now able to examine random forests' effectiveness as a model selection criterion for the macaque data set. Following the workflow defined in figure 1.1 for real data sets, we specified the following binary classification problem using the tools described in section 3.1. SBMs of 4 and 5 communities were estimated on the macaque data set (see figure 3.2

MixNet	Average Density
k = 4	0.2899
k = 5	0.2886

Table 3.1: Average Densities of 4- and 5-Block MixNet Realizations. As reflected in this table, the models fitted to the macaque data set produce graphs of roughly the same density. In this case, the average densities over all realizations used for this experiment are shown. Intuitively, one may expect any classifier to perform poorly once graphs achieve a certain level of similarity with respect to their densities, particularly if one notices that many of our features in section 2.4.1 are closely related to graph density. However, as shown in Caceres et al. (2016) and later in this section, random forests discriminatory power remains quite strong as long as the underlying edge-probabilities remain relatively distinct.

and table 3.1) with 100 graph realizations generated from each model, to be used as a train/test set. Additionally, we also generated another data set in the same manner for use as a further test set. Each graph realization was assigned a label corresponding to the number of communities of the parent SBM. Both the flattened adjacency matrices and the list of network statistics (section 2.4.1) were extracted from the realization data sets as well as the original data as separate feature sets.

To test random forests' ability to select the best generative model for this data set, (where "best" is assumed to be the 4-community model that maximizes the modified ICL criterion), individual forests of 5, 10, 15, ..., 150 trees were constructed. Each forest was fit using 10-fold stratified cross-validation and the classification accuracy scores were averaged over the held-out test sets and again over the additional set of 200 different instance graphs. The classifiers were additionally tasked to select one of the SBMs as a fit for the original data set, receiving a score of 1 if correctly matching the original data set to the 4-block SBM and receiving a 0 otherwise. This process was repeated 100 times for each given number of trees and the final accuracy scores were recorded as averages across these 100 iterations. The entire experiment was conducted first using the flattened adjacency matrices as a feature set, then using network statistics.

3.2.1 Model selection with edge-based classification

Using the flattened adjacency matrices as a feature set, random forests classification accuracy for the SBM realizations increased with the number of trees, leveling out at nearly 100 percent accuracy around 50 trees. The same trend was observed on the additional held-out data set. On the other hand, as a model selection criteria, random forests on a flattened adjacency space typically failed to match



Figure 3.2: Adjacency Matrices of Original Macaque Data with 4- and 5-Block Realizations. Using the methods described in section 3.1, SBMs of 4 and 5 blocks were estimated for the macaque data set. The original adjacency matrix (left) and two realizations of the 4- and 5-block models (top and bottom) are shown.

the original data set with the gold standard 4-community model. As the classifier grew in number of trees, becoming more and more adept at discriminating between the instances of the 4- and 5-block models, the chance of selecting the gold standard model became practically zero.

3.2.2 Model selection with network statistics-based classification

From the perspective of network statistics, random forests did less well in classifying realizations of the 4- and 5-block SBMs but vastly out-performed random forests trained on the flattened adjacency feature space as a model selection criterion. As the number of trees per classifier grew, these models not only became more adept in classifying graph realizations, they also began to consistently match the original data set to the same 4-community model chosen by ICL model selection criteria.

3.2.3 Results

The results for this experiment are summarized in figure 3.3. The green lines, corresponding to "test set 1," represent classification accuracy on the held-out test sets during stratified 10-fold cross-validation. The blue lines for "test set 2" represent classification accuracy on the separate, N = 200

data set. This serves as an additional check on the behavior of random forests by using a "new" data set that was not used in the training/testing phase. Recall that both test sets 1 and 2 consist entirely of instance graphs generated from the 4- and 5- block SBMs. Finally, the red lines represent how often a random forests classifier on a given feature space and number of trees matched the original data set with the gold standard 4-community SBM. In the framework of this experiment, these curves can be interpreted as a rough proportion of the frequency (out of 100) with which the given classifier chooses the optimal model maximizing the ICL criterion in section 3.1.2. Our best random forests model selection criteria was the 140 tree model trained using network statistics, which chose the gold standard model approximately 97 times out of 100. In contrast, the best random forests classifier using flattened adjacency matrices selected the gold standard model for the macaque data set only 28 times out of 100 using 10 trees. Interestingly, strong performance as a model selection criteria does not seem to necessarily translate into strong discriminatory power between graph realizations of the 4- and 5- block SBMs. As seen in the blue and green curves in the lefthand plot, the edge-based classifiers achieve perfect discriminatory power between graph realizations using nTrees > 50. However, network statistics-based classifiers never achieve perfect discriminatory power with respect to the instance graphs, achieving an average accuracy of at most around 87.14% for nTrees = 135.

Our results from chapter 2 would suggest that the use of network statistics as a feature space would lead to stronger classification power when performing model selection. As shown in figure 3.3, this is only half true; random forests modeled on the network statistics feature space do outperform those modeled on the flattened adjacency feature space in terms of model selection, but not in terms of discriminatory power in classifying instances of the candidate models. This suggests that, at least for the macaque data set, raw edge-weights fail to capture some subtleties in the original data that appear to be captured when using our list of network statistics.



Figure 3.3: **Random Forests as a Model Selection Criterion**. *Left*: Classification accuracy versus number of trees for random forests modeled on flattened adjacency matrices of the instance graphs and original data set. *Right*: Classification accuracy versus number of trees for random forests modeled on network statistics (section 2.4.1) of the instance graphs and original data set. For both plots smoothed lines of fit are given, with grey areas representing standard error.

CHAPTER 4

Conclusion and Future Directions

In this thesis, we have outlined a series of experiments that explores random forests' discriminatory power over an extensive range of Stochastic Block Model parameters and structural configurations. To provide further comparison, we implemented these experiments using two different feature spaces, the first taking advantage of the trivial form of connectivity information encoded in raw edge-weights of the graphs - our so-called "flattened adjacency matrix feature space" - the second building off of previous research in network classification by using an extensive list of network statistics. These network statistics can be loosely grouped into measurements related to node degree and graph connectivity, information about shortest paths, and linear algebra concepts using the graphs' adjacency matrices. Overall, classifiers trained on a network statistics feature space not only vastly outperformed those trained on the flattened adjacency feature space, but also generally achieved high rates of classification accuracy. While this is perhaps not a difficult conclusion to comprehend, knowing beforehand that one can expect good discriminatory power in reasonable network classification scenarios by using random forests and network statistics can be extremely helpful to researchers needing fast, easily interpretable results when classifying networks.

This thesis also examines the use of random forests as a generative model selection criterion for real networks data. In other words, we provided an answer to the question "given N candidate stochastic block models for a particular networks data set, is it possible for a random forests classifier trained on a set of instance graphs generated from these N SBMs to select the best fit to a real-world network?" While Caceres et al. (2016) is acknowledged to have also touched on this question, to the knowledge of the authors, ours is the first instance of research exploring this topic in an applied setting and comparing random forests to other accepted model selection methods. As described in Chapter 3, we found that, given a real data set, a random forest classifier trained on a set of graph instances generated from a suite of candidate models will choose the same model selected by the Integrated Classification Likelihood selection criteria over 95% of the time if using the network statistics feature space described previously and a sufficient number of trees. While this is by no means a rigorous way of concluding random forests' generalized viability as a model selection criterion, our work does serve as a promising first step. There are certainly many adequate model selection methods for networks already established and well-proven, however, our method of using random forests not only adds to this area, but also promises a method that is, intuitively speaking, very easy to understand. First, our method uses already well-understood network statistics that can be used to summarize most properties and characteristics of a given graph. Second, we can take advantage of the built-in feature ranking implemented by the random forests algorithm to further understand why a particular generative model is selected for a data set.

Future directions will be aimed at further understanding the behavior of random forests as a model selection criterion. The first step will entail identifying additional real-world networks that are structured in a way that allows the R mixer package to select a viable MixNet model for them. We will then perform experiments similar in nature to those outlined in section 3.2 as a way of verifying the applicability of our method with respect to different data. Assuming model selection with random forests can apply to data sets of all types, we will then delve into exploring the characteristics of random forests as a model selection criteria. One area of exploration is to track which features, if any, are most frequently used by random forests when selecting a model for a data set. Another area is to repeat a series of experiments similar to those performed in Chapter 2. Rather than analyzing the discriminatory power of random forests on synthetic data, we would document random forests' strength as a model selection criteria for many different data sets, and perhaps many different types of generative models. A potential question of interest is whether or not random forests will always select the generative model chosen by a given model selection criterion. Evidence in Chapter 3 suggests that this may heavily depend on the feature set used, in which case we would also like to see if there are any limits to our network statistics feature set, and if so, under what conditions they fail. We also hope to compare our method to a large range of model selection criteria.

BIBLIOGRAPHY

- Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014.
- Barnett, I., Malik, N., Kuijjer, M. L., Mucha, P. J., and Onnela, J.-P. (2016). Feature-based classification of networks. arXiv preprint arXiv:1610.05868.
- Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725.
- Blei, E. A. D. M., Goldenberg, S. E. F. A., and Zheng, E. P. X. A. X. (2007). Statistical network analysis: Models, issues, and new directions.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Caceres, R. S., Weiner, L., Schmidt, M. C., Miller, B. A., and Campbell, W. M. (2016). Model selection framework for graph-based data. arXiv preprint arXiv:1609.04859.
- Canning, J. P., Ingram, E. E., Nowak-Wolff, S., Ortiz, A. M., Ahmed, N. K., Rossi, R. A., Schmitt, K. R., and Soundarajan, S. (2017). Network classification and categorization. *arXiv preprint* arXiv:1709.04481.
- Centola, D. (2010). The spread of behavior in an online social network experiment. *science*, 329(5996):1194–1197.
- Clauset, A. (2013). Network Analysis and Modeling, CSCI 5352, Lecture 16. Network Analysis and Modeling CSCI 5352, Fall 2013. Santa Fe Institute.
- Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and computing*, 18(2):173–183.
- Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.
- Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.
- Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
- Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. Social networks, 5(2):109–137.
- LAPACK, Ambroise, C., Latouche, P., Grasseau, G., Hoebeke, M., Miele, V., and Picard, F. (2015). Mixer v1.8.
- Li, G., Semerci, M., Yener, B., and Zaki, M. J. (2012). Effective graph classification based on topological and label attributes. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4):265–283.

- Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878.
- Newman, M. (2010). Networks: an introduction. Oxford university press.
- Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, 67(2):026126.
- Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.
- Peixoto, T. P. (2015). Model selection and hypothesis testing for large-scale network models with overlapping groups. *Physical Review X*, 5(1):011033.
- Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.
- Singh, S. S., Khundrakpam, B., Reid, A. T., Lewis, J. D., Evans, A. C., Ishrat, R., Sharma, B. I., and Singh, R. B. (2016). Scaling in topological properties of brain networks. *Scientific reports*, 6:24926.
- Stam, C. J. (2014). Modern network science of neurological disorders. *Nature Reviews Neuroscience*, 15(10):683.
- Waugh, A. S., Pei, L., Fowler, J. H., Mucha, P. J., and Porter, M. A. (2009). Party polarization in congress: A network science approach.
- Yanardag, P. and Vishwanathan, S. (2015). Deep graph kernels. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1365– 1374. ACM.