STATISTICAL ANALYSIS OF NETWORKS WITH COMMUNITY STRUCTURE AND BOOTSTRAP METHODS FOR BIG DATA

BY

SRIJAN SENGUPTA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Statistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Professor Yuguo Chen, Co-Chair
Associate Professor Xiaofeng Shao, Co-Chair
Professor John Marden
Professor Douglas Simpson

ABSTRACT

This dissertation is divided into two parts, concerning two areas of statistical methodology. The first part of this dissertation concerns statistical analysis of networks with community structure. The second part of this dissertation concerns bootstrap methods for big data.

Statistical analysis of networks with community structure:
Networks are ubiquitous in today's world — network data appears from varied fields such as scientific studies, sociology, technology, social media and the Internet, to name a few. An interesting aspect of many real-world networks is the presence of community structure and the problem of detecting this community structure.

In the first chapter, we consider heterogeneous networks which seems to have not been considered in the statistical community detection literature. We propose a blockmodel for heterogeneous networks with community structure, and introduce a heterogeneous spectral clustering algorithm for community detection in heterogeneous networks. Theoretical properties of the clustering algorithm under the proposed model are studied, along with simulation study and data analysis.

A network feature that is closely associated with community structure is the *popularity*[1] of nodes in different communities. Neither the classical stochastic blockmodel nor its degree-corrected extension can satisfactorily capture the dynamics of node popularity. In the second chapter, we propose a *popularity-adjusted blockmodel* for flexible modeling of node popularity. We establish consistency of likelihood modularity for community detection under the proposed model, and illustrate the improved empirical insights that can be gained through this methodology by analyzing the political blogs network

[1] Popularity is defined as the number of edges between a specific node and a specific community.

and the British MP network, as well as in simulation studies.

Bootstrap methods for big data:
Resampling methods provide a powerful method of evaluating the precision of a wide variety of statistical inference methods. The complexity and massive size of big data makes it infeasible to apply traditional resampling methods for big data.

In the first chapter, we consider the problem of resampling for irregularly spaced dependent data. Traditional block-based resampling or subsampling schemes for stationary data are difficult to implement when the data are irregularly spaced, as it takes careful programming effort to partition the sampling region into complete and incomplete blocks. We develop a resampling method called Dependent Random Weighting (DRW) for irregularly spaced dependent data, where instead of using blocks we use random weights to resample the data. By allowing the random weights to be dependent, the dependency structure of the data can be preserved in the resamples. We study the theoretical properties of this resampling methods as well as its numerical performance in simulations.

In the second chapter, we consider the problem of resampling in massive data, where traditional methods like bootstrap (for independent data) or moving block bootstrap (for dependent data) can be computationally infeasible since each resample has effective size of the same order as the sample. We develop a new resampling method called subsampled double bootstrap (SDB) for both independent and stationary data. SDB works by choosing small random subsets of the massive data, and then constructing a single resample from that subset using bootstrap (for independent data) or moving block bootstrap (for stationary data). We study theoretical properties of SDB as well as its numerical performance in simulated data and real data.

Extending the underlying ideas of the second chapter, we introduce two new resampling strategies for big data in Chapter 3. The first strategy is called aggregation of little bootstraps or ALB, a generalized resampling technique that includes the SDB as a special case. The second strategy is called subsampled residual bootstrap or SRB, a fast version of residual bootstrap intended for massive regression models. We study both methods through simulations.

This thesis is dedicated to two beautiful, brilliant, spirited ladies:
to my wife, Swarnali Sanyal, for being an amazing partner and friend, a steadfast source of support, and a wonderfully patient sounding board, and to my little niece, Roopkotha Guha, for being a source of absolute joy, whose growing up is the biggest thing I regret missing while working on this thesis.

ACKNOWLEDGMENTS

I would like to gratefully acknowledge my advisors, Professor Yuguo Chen and Professor Xiaofeng Shao, for their mentorship, guidance, and support during the preparation of this thesis. I am grateful to Professor Douglas Simpson and Professor John Marden for being in my dissertation committee and for helpful suggestions.

I would like to thank the Department of Statistics at the University of Illinois at Urbana-Champaign for providing me the opportunity and the resources to pursue my doctoral degree. I am privileged to have had the opportunity to interact with many smart, friendly peers in the department and the university.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Statistical analysis of networks with community structure

Many complex systems in today's world consist, at an abstract level, of *agents* who *interact* with one another. This general agent-interaction framework describes many interesting and important systems, such as social interpersonal systems (Milgram, 1967), protein interaction systems (Gavin et al., 2002), power grids (Watts and Strogatz, 1998), and the World Wide Web (Huberman and Adamic, 1999), to name a few. Networks provide a convenient and unified way of representing such systems arising from diverse applications. It is therefore important to develop methodology for network data, and accordingly, the science of network data has received attention from scientists in various academic fields. A holistic introduction to the interdisciplinary study of networks can be found in Newman (2010). Statistically oriented overview of networks can be found in Goldenberg et al. (2010) and Kolaczyk (2009).

The early approach to network modeling, the random graph model by Erdös and Rényi (1959), assumed that all agents behave in identical fashion. The observed dissimilarity in agent behavior was assigned to random fluctuations. This explanation might not always be appropriate, particularly when the network displays structured dissimilarities in agent behavior. At the other end of the modeling spectrum, one might wish to capture the observed variation in agent behavior by assigning a separate model to each individual agent. However, this might be impractical for networks beyond a certain size, and also unnecessary.

Real world networks often exhibit a *patterned dissimilarity* that lies somewhere between completely identical agent behavior and completely unequal agent behavior. The agents are often found to cluster into groups or communities that display similar behavior, while agents from different communities behave differently. The identification of this network structure, called *community detection*, is an important problem in network analysis. Community detection has important real-world interpretation; these communities often turn out to be groups of agents which share common properties and/or play similar roles within the network. For example, in Jonsson et al. (2006), the communities in a protein interaction network turned out to be functional groups (proteins having the same or similar function) — this conclusion has important implications for cancer research. Fortunato (2010) provides a multidisciplinary exposition on community detection in networks.

In Chapter 2, we consider heterogeneous networks which seems to have not been considered in the statistical community detection literaure. Many real-world systems consist of several *types* of entities, and heterogeneous networks are required to represent such systems. However, the current statistical toolbox for network data can only deal with homogeneous networks, where all nodes are supposed to be of the same type. This article introduces a statistical framework for community detection in heterogeneous networks. For modeling heterogeneous networks, we propose heterogeneous versions of both the classical stochastic blockmodel and the degree-corrected blockmodel. For community detection, we formulate heterogeneous versions of standard spectral clustering and regularized spectral clustering. We also demonstrate the theoretical accuracy of the proposed heterogeneous methods for networks generated from the proposed heterogeneous models. Our simulations establish the superiority of proposed heterogeneous methods over existing homogeneous methods in finite networks generated from the models. An analysis of the DBLP four-area data demonstrates the improved accuracy of the heterogeneous method over the homogeneous method in identifying research areas for authors.

In ongoing work in Chapter 3, we consider the problem of modeling of popularity of nodes across communities, which is network feature closely associated with community structure. In this chapter we introduce a new random graph model, called popularity-adjusted blockmodel (PABM, hereafter) for networks with community structure. Our model incorporates popularity parameters for each node that can take different values for different communities. The profile likelihood for this model is formulated as a modularity function for community detection. We compare the methodology with existing methodology using simulated and real networks.

1.2 Bootstrap methods for big data

Given a dataset, the primary task of data analysis is usually to perform some kind of statistical inference on this data. This inference task can be the estimation of a parameter of interest, the test of a statistical hypothesis, and so on. A secondary task that is inextricably associated with any statistical inference method, is to assess the risk or precision associated with that inference. For example, suppose the goal of data analysis is the estimation of the population mean, and this primary task of estimation is performed using the sample mean. A natural next step would be the task of evaluating the degree of precision of the sample mean as an estimator of the population.

A measure of the precision of an inference method usually refers to the unknown sampling distribution of a statistic, the underlying distribution from which the observed sample mean can be visualized as a random draw. Often we have theoretical ideas about the sampling distributions in an asymptotic sense, but very little empirical information, since all we observe is a single random draw from this distribution. If it was possible to repeatedly draw random samples from this unknown distribution, we could have more empirical information on it. However, this is typically not feasible in practice. An approximate computational method is resample — consider the sample as a proxy for the population, and draw repeated resamples from the sample. Resampling methods provide a general and powerful method of evaluating the precision of a wide variety of statistical inference methods.

In chapter 4, we consider the problem of resampling for irregularly spaced dependent data. Traditional block-based resampling or subsampling schemes for stationary data are difficult to implement when the data are irregularly spaced, as it takes careful programming effort to partition the sampling region into complete and incomplete blocks. We propose a new resampling method, the dependent random weighting, for both time series and random fields. The method is a generalization of the traditional random weighting in that the weights are made to be temporally or spatially dependent and are adaptive to the configuration of the data. Unlike the block-based bootstrap or subsampling methods, the dependent random weighting can be used for irregularly spaced time series and spatial data without any implementational difficulty. Consistency of the distribution approximation is shown for both equally and unequally spaced time series. Simulation studies illustrate the finite sample performance of the dependent random weighting in comparison with the existing counterparts for both one dimensional and two dimensional irregularly spaced data.

In chapter 5, we consider the problem of resampling in massive data. The bootstrap is a popular and powerful method for estimating precision of estimators and inferential methods. However, for massive datasets which are increasingly prevalent, the bootstrap becomes prohibitively costly in computation and its feasibility is questionable even with modern parallel computing platforms. Recently Kleiner et al. (2014) proposed a method called BLB (Bag of Little Bootstraps) for massive data which is more computationally scalable with little sacrifice of statistical accuracy. Building on BLB and the idea of fast double bootstrap, we propose a new resampling method, the subsampled double bootstrap, for both independent data and time series data. In theory, we establish the consistency of our subsampled double bootstrap under mild conditions for both independent and dependent cases. Methodologically, the subsampled double bootstrap is superior to BLB in terms of running time, more sample coverage and automatic implementation with less tuning parameters for a given time budget. Its advantage relative to BLB and bootstrap is also demonstrated in numerical experiments and data illustrations.

In chapter 6, we continue studying bootstrap methods for big data applications. Extending the underlying idea of scaling down the effective sample size, we introduce two new resampling strategies for big data. The first strategy is called aggregation of little bootstraps or ALB, a generalized resampling technique that includes the SDB as a special case. Instead of taking the mean of estimates from different subsets (as in BLB), we *aggregate* all resampled roots $\{T_n^{**,s,r}\}_{s=1,\dots,S,r=1,\dots,R}$ into a single ensemble, and compute the precision measure from the empirical cdf of this ensemble. The second strategy is called subsampled residual bootstrap or SRB, a fast version of residual bootstrap intended for massive regression models. Instead of a full-size resampling of regression residuals, we construct a subsample and use appropriate scaling adjustments to obtain a fast alternative to classical residual bootstrap. We study both methods through simulations.

CHAPTER 2

SPECTRAL CLUSTERING IN HETEROGENEOUS NETWORKS

2.1 Introduction

Many complex systems in today's world consist, at an abstract level, of *agents* who *interact* with one another. This general agent-interaction framework describes many interesting and important systems, such as social interpersonal systems (Milgram, 1967), protein interaction systems (Gavin et al., 2002), power grids (Watts and Strogatz, 1998), and the World Wide Web (Huberman and Adamic, 1999), to name a few. Networks provide a convenient and unified way of representing such systems arising from diverse applications. It is therefore important to develop methodology for network data, and accordingly, the science of network data has received attention from scientists in various academic fields. A holistic introduction to the interdisciplinary study of networks can be found in Newman (2010). Statistically oriented overview of networks can be found in Goldenberg et al. (2010) and Kolaczyk (2009).

The early approach to network modeling, the random graph model by Erdös and Rényi (1959), assumed that all agents behave in identical fashion. The observed dissimilarity in agent behavior was assigned to random fluctuations. This explanation might not always be appropriate, particularly when the network displays structured dissimilarities in agent behavior. At the other end of the modeling spectrum, one might wish to capture the observed variation in agent behavior by assigning a separate model to each individual agent. However, this might be impractical for networks beyond a certain size, and also unnecessary.

Real world networks often exhibit a *patterned dissimilarity* that lies somewhere between completely identical agent behavior and completely unequal agent behavior. The agents are often found to cluster into groups or communities that display similar behavior, while agents from different communities behave differently. The identification of this network structure, called *community detection*, is an important problem in network analysis. Community detection has important real-world interpretation; these communities often turn out to be groups of agents which share common properties and/or play similar roles within the network. For example, in Jonsson et al. (2006), the communities in a protein interaction network turned out to be functional groups (proteins having the same or similar function) — this conclusion has important implications for cancer research. Fortunato (2010) provides a multidisciplinary exposition on community detection in networks.

The currently available methodologies for network data usually consider networks to be *homogeneous*, that is, the nodes in the network represent objects of the same type and all links in the network represent the same type of relation. For example, a friendship network, say Facebook, has nodes representing persons or users, and links representing friendship between users. However, many real-world systems are actually *heterogeneous*, in the sense that there are different *types* of agents, and various kinds of interactions in the system. Consequently, networks representing such systems are also heterogeneous, where several types of nodes and several types of links exist in the same network. Typically, for each node or link it is known what the type is, and a heterogeneous network contains this type information. For example, in Facebook, nodes can represent various types of entities like users, events, groups, celebrity pages, photos, and so on. Accordingly, there can be various types of links: friendship link between two users, membership link between users and groups, fan (or *like*) link between users and celebrities, attendance link between users and events, *tag* link between a photo and an user, and so on. The homogeneous 'friendship network' representation, that was mentioned earlier, effectively represents only a sub-system of this system, consisting only of 'user' nodes and 'friendship' links.

To analyze a heterogeneous network using the current toolbox of homogeneous methods and homogeneous models, there are two options — either consider a homogeneous sub-network of the original network, or treat the heterogeneous network as a homogeneous network, suppressing the type information available in the data. In the first approach, there is loss of useful information. In the second approach the results might be meaningless as nodes of different types are grouped into the same community, or the procedure might not work well due to the presence of different types of nodes.

For example, consider a heterogeneous Facebook network consisting of two types of nodes — users and events, and two kinds of links: user-user or *friendship* links, and user-event or *attendance* links. Suppose network data in this form is available for users and events corresponding to 10 universities, and the problem of interest is to assign users to their universities using a clustering procedure. Using the first option, one must carry out the analysis based on the user-user network only, dumping the event nodes and user-event links. In this context the dumped data can be quite important in predicting university affiliation. Using the second option, one treats the entire network as a homogeneous network and carries out a clustering of both users and events. However, users and events behave in very different ways, and the clustering algorithm might not work well since it is trying to cluster these different entities into the same clusters by comparing their behavior. Using K-means intuition, the 'distance' between an user and an event, both affiliated to the same university, might be too large.

Thus, community detection in heterogeneous network data cannot be satisfactorily carried out by applying homogeneous models and methodologies. A preferable approach is to have a procedure that uses the entire heterogeneous information, identifies the fact that users and events are different types of entities, and clusters nodes from different types *separately but simultaneously* into 10 user clusters and 10 event clusters. Since this procedure compares events to events and users to users, the clustering should work much better.

Heterogeneous networks have begun to receive attention from various scientific communities, particularly the computer science research community (Sun and Han, 2012). This chapter provides a statistical framework to deal with heterogeneous network data by extending the existing homogeneous framework. For modeling heterogeneous networks, we propose heterogeneous versions of the classical stochastic blockmodel and the degree-corrected blockmodel recently proposed by Karrer and Newman (2011). For community detection in heterogeneous networks, we formulate heterogeneous versions of standard spectral clustering and regularized spectral clustering. We also demonstrate the theoretical accuracy of the proposed heterogeneous methods for networks generated from the proposed heterogeneous models in the asymptotic framework of Qin and Rohe (2013).

As a real-world application of our methods, we implement our algorithm on a large bibliographical network from DBLP with the objective of identifying research area of authors. Under the existing homogeneous paradigm, the natural choice of network would be the co-authorship network with authors as nodes. We find that homogeneous clustering applied on the co-authorship network performs rather poorly, with an accuracy comparable to random assignment. However, interpreting the bibliographical network as a heterogeneous network (with authors, papers and conferences treated as different types of nodes), the heterogeneous clustering method performs quite accurate community detection.

The rest of the chapter is organized as follows. Section 2.2 outlines basic graph theoretical notation that is used throughout the chapter. Section 2.3 reviews existing homogeneous blockmodels and introduces heterogeneous versions of these models. Section 2.4 discusses the standard and regularized spectral clustering algorithms and presents modified versions of these algorithms that are appropriate for heterogeneous networks. Section 2.5 provides a brief outline of the asymptotic framework of Qin and Rohe (2013), and demonstrates the asymptotic accuracy of the heterogeneous algorithms under the heterogeneous models, using this framework. Section 2.6 presents simulation studies demonstrating various circumstances under which the heterogeneous methods can provide significant improvements in clustering accuracy over the homogeneous methods. Section 2.7 presents a real-life example of the superiority of the heterogeneous method over the homogeneous method, using the DBLP four-area dataset. The chapter concludes with the discussion in Section 3.7.

2.2 Graph Theoretic Notation

Mathematically a network is represented as a *graph* $G = (V, E)$ consisting of two types of elements, namely *nodes* (or *vertices*) that comprise the set V, and *links* (or *edges*) that make up the set E. Every link has two endpoints in the set of nodes, and is said to *connect* or *join* the two nodes. The two endpoints of a link are also said to be *adjacent* to each other, or *neighbors*. An unweighted, undirected graph containing no self-loops or multiple edges is called a *simple graph*. The *degree* d_v of a node v in a graph G is the number of nodes adjacent to v. A *degree sequence* is a list of degrees of a graph in non-increasing order (e.g., $d_1 \geq d_2 \geq \cdots \geq d_n$).

An *adjacency matrix* $\mathbf{A}$ is often used to represent a graph: for a graph with N nodes, it is an N-by-N matrix whose (i, j)-th entry gives the number of links from the i-th node to the j-th node. This chapter covers simple graphs only, and hence the adjacency matrix is symmetric, consists only of 0's and 1's, and all its diagonal entries are zero.

The *Graph Laplacian* $\mathbf{L}$ is a matrix frequently used in network analysis. There are several ways of defining the Laplacian; in this chapter it is defined as

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{2.1}$$

where $\mathbf{A}$ is the adjacency matrix and $\mathbf{D}$ is the degree matrix (i.e., a diagonal matrix whose i^{th} diagonal element is the degree of node i). This version of the Laplacian is often referred to as the symmetric normalised Laplacian, but this chapter will simply refer to this as the Laplacian.

2.3 Stochastic Blockmodel and Degree Corrected Blockmodel for Heterogeneous Networks

Lorrain and White (1971) were the first to introduce blockmodels, in association with the deterministic concept of *structural equivalence*, where two nodes of a network are considered equivalent if they have the same set of neighbors. Holland et al. (1983) and Fienberg et al. (1985) generalized this equivalence concept to a probabilistic setting, calling it *stochastic equivalence.* In contrast to structural equivalence which is defined with respect to the observed network itself, stochastic equivalence is defined with respect to the conceptual model that generates the observed network.

Definition Two nodes in a network are said to be *stochastically equivalent* if the probability of any event pertaining to the network remains unchanged by exchanging the node labels.

For a homogeneous network, two nodes (say, 1 and 2) are stochastically equivalent according to this definition if they have the same probability of being linked to *any* third node (say 3).

2.3.1 Homogeneous model

Consider a simple graph $G = (V, E)$ with N nodes, and let $\mathbf{A}$ be its adjacency matrix. Note that $\mathbf{A}$ is a symmetric 0-1 matrix, and its diagonal entries are all zero. Under the K-block stochastic blockmodel, there are K blocks, and each node belongs to one of these blocks. Let $\mathbf{M}$ denote the N-by-K block membership matrix with $\mathbf{M}(i,k) = 1$ if node i is in the k^{th} block, and $\mathbf{M}(i,k) = 0$ otherwise. Then for $i < j$, under the stochastic blockmodel (SBM) $A(i,j)$ are Bernoulli random variables, with

$$E[\mathbf{A}(i,j) \mid \mathbf{M}] = \mathbf{M}(i,\cdot)\mathbf{P}\mathbf{M}(j,\cdot)', \tag{2.2}$$

where $\mathbf{P}$ is the K-by-K matrix of link probabilities, that is, $\mathbf{P}(a,b)$ represents the probability that a node in block a is linked to another node in block b. Edges are conditionally independent given the membership matrix $\mathbf{M}$.

Model (3.2) essentially means that if nodes i and i' come from the same block, i.e., $\mathbf{M}(i,\cdot) = \mathbf{M}(i',\cdot)$, then they are stochastically equivalent; for any j different from i and i', the links $\mathbf{A}(i,j)$ and $\mathbf{A}(i',j)$ are exchangeable — hence, exchanging the node labels of i and i' will make no difference to the probability of any event in the network.

Stochastic equivalence theorizes that two nodes in the same block have identical degree distributions. This can be an unrealistic assumption for many empirical networks. The degree-corrected blockmodel (DCBM) proposed by Karrer and Newman (2011) adds degree scaling parameters θ_i for each node to allow for a broad degree distribution. Then for $i < j$, under the DCBM $A(i,j)$ are Bernoulli random variables, with

$$E[\mathbf{A}(i,j) \mid \mathbf{M}] = \mathbf{\Theta}(i,\cdot)\mathbf{MPM'}\mathbf{\Theta}(\cdot,j) \tag{2.3}$$

where $\mathbf{\Theta}$ is an N-by-N diagonal matrix with $\mathbf{\Theta}(i,i) = \theta_i$, the degree parameter of the i^{th} node, and all other parameters have the same meaning as (3.2).

Note that the SBM is a special case of the DCBM when all nodes in the same block have equal value of the θ_i, the degree parameter.

2.3.2 Heterogeneous model

A heterogeneous network has nodes of several different *types*. Nodes of different types are fundamentally different in their role in the network. Similarly, the links in the network are also of different kinds, depending upon the types of the nodes they link. Therefore, a blockmodel for heterogeneous networks should allow the link probabilities to change not only by block but also by node type. We propose the following model for accommodating this.

Consider a K-block heterogeneous network with N nodes of T different types. We divide each block into T sub-blocks for different types of nodes such that each type-block combination is represented by a separate sub-block. Let $\mathbf{M}$ represent the N-by-TK *sub-block* membership matrix, with $\mathbf{M}(i, t\times k) = 1$ if node i is of the t^{th} type and belongs to the k^{th} block, and $\mathbf{M}(i, t\times k) = 0$ otherwise, for $t = 1,\dots,T$ and $k = 1,\dots,K$. Let $\mathbf{P}$ be the TK-by-TK matrix of link probabilities. Then $\mathbf{P}$ has the following structure:

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1T} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2T} \\ & & \dots & \\ \mathbf{P}_{T1} & \mathbf{P}_{T2} & \dots & \mathbf{P}_{TT} \end{pmatrix},$$

where $\mathbf{P}_{st}$ is the K-by-K matrix of probabilities for type s-type t links. Thus, $\mathbf{P}_{st}(a,b)$ represents the probability that a node of the s^{th} type and belonging to block a is linked to another node of the t^{th} type and belonging to block b.

As before, for $i < j$, $\mathbf{A}(i,j)$ are Bernoulli random variables, with

$$E[\mathbf{A}(i,j) \mid \mathbf{M}] = \mathbf{M}(i,\cdot)\mathbf{P}\mathbf{M}(j,\cdot)' \tag{2.4}$$

for the heterogeneous stochastic blockmodel (Het-SBM) and

$$E[\mathbf{A}(i,j) \mid \mathbf{M}] = \mathbf{\Theta}(i,\cdot)\mathbf{M}\mathbf{P}\mathbf{M}'\mathbf{\Theta}(\cdot,j) \tag{2.5}$$

for the heterogeneous degree-corrected blockmodel (Het-DCBM).

This complicated representation of link probabilities is necessary, because link probabilities vary not only by block, but also by type; $\mathbf{P}_{st}(a,b)$ varies not only with a and b but also with s and t.

For illustration, consider a toy example for Het-SBM with number of types $T = 2$, the number of blocks $K = 3$ and the number of nodes $N = 30$, with 5 type 1 nodes and 5 type 2 nodes in each block, and the link probability matrix as follows:

$$\mathbf{P} = \begin{pmatrix} 0.75 & 0.25 & 0.25 & 0.90 & 0.00 & 0.00 \\ 0.25 & 0.75 & 0.25 & 0.00 & 0.90 & 0.00 \\ 0.25 & 0.25 & 0.75 & 0.00 & 0.00 & 0.90 \\ & & & & & \\ 0.90 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.90 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.90 & 0.00 & 0.00 & 0.00 \end{pmatrix}.$$

Link probabilities vary prominently across blocks as well as types in this model. Type 1 nodes are strongly homophilic (intra-community links are much more likely than inter-community links), while type 2 nodes are not linked among themselves at all. The type 1-type 2 links have even stronger homophily; type 1 and type 2 nodes belonging to the same block are very likely to be connected, while inter-block, inter-type links are not present. This model is an exaggerated representation of the user-event heterogeneous Facebook system mentioned in the introduction.

In Figure 2.1, type 1 nodes and type 2 nodes are clearly different in their roles in the network, nevertheless they form close-knit communities. Visually, it appears that community structure is stronger in the entire network, compared to the homogeneous type 1-type 1 subnetwork. This toy example gives a visual intuition of how community discovery might be more accurate in the presence of heterogeneous information.

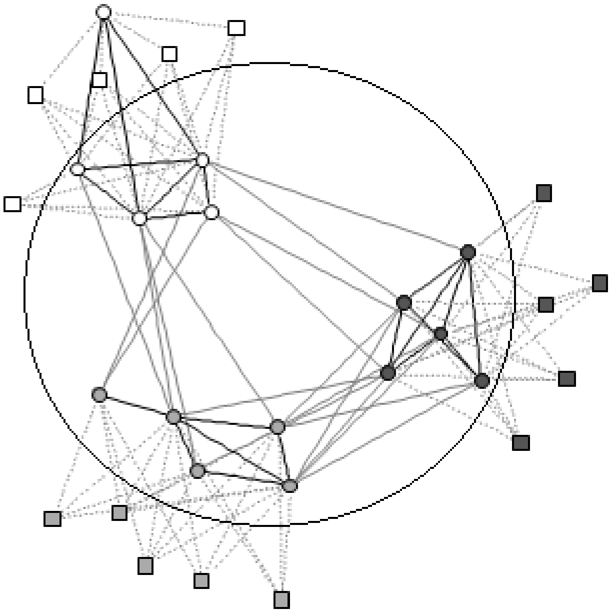


Figure 2.1: Sample heterogeneous network with $T = 2, K = 3$ and $N = 30$, with 5 type 1 nodes (circles) and 5 type 2 nodes (squares) in each block. Solid lines (black for intra-block, gray for inter-block) represent type 1-type 1 links, while dotted gray lines represent type 1-type 2 links. The homogeneous type 1-type 1 subnetwork is approximately enclosed in the circle.

2.4 Spectral Clustering and Regularized Spectral Clustering

2.4.1 Homogeneous clustering

Consider a homogeneous network with N nodes and let $\mathbf{A}$ be its adjacency matrix. Assuming a correctly specified K-block blockmodel structure for this network, the standard spectral clustering algorithm assigns the N nodes to K clusters in the following steps.

Homogeneous Spectral Clustering Algorithm (Hom-SC)

1. Given the adjacency matrix $\mathbf{A}$, calculate the graph Laplacian $\mathbf{L}$ by (2.1).

2. Find orthonormal eigenvectors $\mathbf{X}_1, \ldots, \mathbf{X}_K$ corresponding to the K eigenvalues of $\mathbf{L}$ that are largest in absolute value. Put them into the N-by-K matrix $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_K]$.

3. Carry out a K-means clustering with the N rows of matrix $\mathbf{X}$, creating a K-partition of the index set $\{1, \ldots, N\}$.

4. Assign the nodes to clusters in accordance to the clustering obtained in step 3, i.e., assign the i^{th} node to the k^{th} cluster if the i^{th} row got assigned to the k^{th} cluster in step 3.

Recent work by Amini et al. (2013) and Jin (2012) demonstrate that homogeneous spectral clustering does not work very well in sparse homogeneous networks with wide degree distribution. Chaudhuri et al. (2012) proposed a regularized version of the graph Laplacian for sparse networks and it was shown by Qin and Rohe (2013) that a normalized variant of spectral clustering on this regularized Laplacian has superior theoretical properties under the degree corrected stochastic blockmodel. In this context we would also like to mention Joseph and Yu (2013) for their in-depth analysis of the performance of a slightly different version of regularized spectral clustering.

For a regularizer $\tau \geq 0$, define the regularized degree matrix $\mathbf{D}_\tau = \mathbf{D} + \tau\mathbf{I}$ and the regularized graph Laplacian

$$\mathbf{L}_\tau = \mathbf{D}_\tau^{-1/2}\mathbf{A}\mathbf{D}_\tau^{-1/2}. \tag{2.6}$$

Following the recommendation of Qin and Rohe (2013), in this paper we set τ equal to the average node degree of the network, in all applications of regularized spectral clustering (homogeneous and heterogeneous) for simulations as well as data analysis. The regularized spectral clustering algorithm assigns the N nodes to K clusters in the following steps.

Homogeneous Regularized Spectral Clustering Algorithm (Hom-RSC)

1. Given the adjacency matrix $\mathbf{A}$ and regularizer $\tau \geq 0$, calculate regularized graph Laplacian $\mathbf{L}_\tau$ by (2.6).

2. Find orthonormal eigenvectors $\mathbf{X}_1, \ldots, \mathbf{X}_K$ corresponding to the K eigenvalues of $\mathbf{L}_\tau$ that are largest in absolute value. Put them into the N-by-K matrix

$\mathbf{X}_\tau = [\mathbf{X}_1, \ldots, \mathbf{X}_K]$.

3. Normalize each row of $\mathbf{X}_\tau$ to have unit norm, forming the N-by-K matrix $\mathbf{X}_\tau^*$ given by $\mathbf{X}_\tau^*(i,j) = \mathbf{X}_\tau(i,j) \Big/ \sqrt{\sum_j \mathbf{X}_\tau(i,j)^2}$.

4. Carry out a K-means clustering with the rows of matrix $\mathbf{X}_\tau^*$, creating a K-partition of the index set $\{1, \ldots, N\}$.

5. Assign the nodes to clusters in accordance to the clustering obtained in step 4, i.e., assign the i^{th} node to the k^{th} cluster if the i^{th} row got assigned to the k^{th} cluster in step 4.

2.4.2 Heterogeneous clustering

For a T-type heterogeneous network, there are TK clusters (since each type-block combination represents a cluster), but for each node, the type information is already known. So essentially there are T cluster assignment problems — to assign the n_1 type 1 nodes into K clusters, the n_2 type 2 nodes into K separate clusters, and so on. This can be achieved by carrying out T *simultaneous but separate* K-means clustering procedures. We now present heterogeneous versions of the Hom-SC and Hom-RSC algorithms based on this idea.

Heterogeneous Spectral Clustering Algorithm (Het-SC)

1. Given the adjacency matrix $\mathbf{A}$, calculate the graph Laplacian $\mathbf{L}$ by (2.1).

2. Find orthogonal eigenvectors $\mathbf{X}_1, \ldots, \mathbf{X}_{TK}$ corresponding to the TK eigenvalues of $\mathbf{L}$ that are largest in absolute value. Put them into the N-by-TK matrix $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_{TK}]$.

3. For each $t = 1, \ldots, T$, select the n_t rows of $\mathbf{X}$ that correspond to nodes of type t, and carry out separate K-means clustering for each selection, creating a TK-partition of the index set $\{1, \ldots, N\}$.

4. Assign the nodes to clusters in accordance to the clustering obtained in step 3, i.e., assign the i^{th} node to the r^{th} cluster if the i^{th} row got assigned to the r^{th} cluster in step 3.

Heterogeneous Regularized Spectral Clustering Algorithm (Het-RSC)

1. Given the adjacency matrix $\mathbf{A}$ and regularizer $\tau \geq 0$, calculate the regularized graph Laplacian $\mathbf{L}_\tau$ by (2.6).

2. Find orthogonal eigenvectors $\mathbf{X}_1, \ldots, \mathbf{X}_{TK}$ corresponding to the TK eigenvalues of $\mathbf{L}_\tau$ that are largest in absolute value. Put them into the N-by-TK matrix $\mathbf{X}_\tau = [\mathbf{X}_1, \ldots, \mathbf{X}_{TK}]$.

3. Normalize each row of $\mathbf{X}_\tau$ to have unit norm, forming the N-by-TK matrix $\mathbf{X}^*_\tau$ given by $\mathbf{X}^*_\tau(i,j) = \mathbf{X}_\tau(i,j) \Big/ \sqrt{\sum_j \mathbf{X}_\tau(i,j)^2}$.

4. For each $t = 1, \dots, T$, select the n_t rows of $\mathbf{X}^*_\tau$ that correspond to nodes of type t, and carry out separate K-means clustering for each selection, creating a TK-partition of the index set $\{1, \dots, N\}$.

5. Assign the nodes to clusters in accordance to the clustering obtained in step 4, i.e., assign the i^{th} node to the r^{th} cluster if the i^{th} row got assigned to the r^{th} cluster in step 4.

In the next section we provide theoretical justification of why the Het-RSC algorithm works under the Het-DCBM model and the Het-SC algorithm works under the Het-SBM model.

2.5 Convergence of Heterogenous Spectral Clustering

This section outlines an asymptotic theory for the convergence of the Hom-RSC algorithm under the Hom-DCBM, and proposes a similar result for the Het-RSC algorithm under the Het-DCBM. Convergence of the Het-SC algorithm under the Het-SBM follows as a special case. The interested reader is directed to Qin and Rohe (2013) for technical details for the homogeneous case.

For the Hom-DCBM (3.3), define $\mathcal{A} = \mathbf{\Theta}\mathbf{M}\mathbf{P}\mathbf{M}'\mathbf{\Theta}$ and let $\mathcal{D}$ be the diagonal matrix of expected degrees, i.e., $\mathcal{D}(i,i) = \sum_j \mathcal{A}(i,j)$. Define $\mathcal{D}_\tau = \mathcal{D} + \tau\mathbf{I}$ and let $\mathcal{L}_\tau = \mathcal{D}_\tau^{-1/2}\mathcal{A}\mathcal{D}_\tau^{-1/2}$ be the population version of the regularized graph Laplacian (2.6). Under a K-block Hom-DCBM, $\mathcal{L}_\tau$ has exactly K non-zero eigenvalues. Let $\mathcal{X}_\tau$ be the N-by-K matrix of the corresponding eigenvectors. Finally, $\mathcal{X}^*_\tau$ is the row-normalized version of $\mathcal{X}_\tau$. The main idea is to interpret the clustering algorithm as an estimation procedure for $\mathbf{X}^*$ with $\mathcal{X}^*_\tau$ as the parameter.

Let $\delta = \min_{i=1,\dots,N} \mathcal{D}(i,i)$ be the minimum expected degree, and let λ be the magnitude of the smallest non-zero eigenvalue of $\mathcal{L}_\tau$ in magnitude. Let $\gamma = \min_{i=1,\dots,N}\{\min\{||\mathbf{X}_\tau(i,\cdot)||_2, ||\mathcal{X}_\tau(i,\cdot)||_2\}$ be the length of the shortest row in $\mathcal{X}_\tau$ and $\mathbf{X}_\tau$, where $||x||_2$ represents the L^2 norm of the vector x. Assume that for some $\epsilon > 0$ and sufficiently large N,

$$(A1)\ \delta + \tau > 3\log(4N/\epsilon) \quad \text{and} \quad (A2)\ \lambda \geq 8\sqrt{\frac{3K\log(4N/\epsilon)}{\delta+\tau}}.$$

Theorem 4.2 of Qin and Rohe (2013) states: when $(A1)$ and $(A2)$ hold, then

$$||\mathbf{X}_\tau - \mathcal{X}_\tau \mathbf{O}||_F \leq c_0 \frac{1}{\lambda} \sqrt{\frac{K \log(4N/\epsilon)}{\delta + \tau}} \tag{2.7}$$

and

$$||\mathbf{X}_\tau^* - \mathcal{X}_\tau^* \mathbf{O}||_F \leq c_0 \frac{1}{\gamma\lambda} \sqrt{\frac{K \log(4N/\epsilon)}{\delta + \tau}} \tag{2.8}$$

for some constant c_0 with probability at least $1 - \epsilon$, where $\mathbf{O}$ represents an orthonormal rotation, and $|| \cdot ||_F$ denotes the Frobenius norm of a matrix, defined as $||\mathbf{B}||_F = \sqrt{\sum_i \sum_j |\mathbf{B}(i,j)|^2}$.

The next step is to translate this accuracy in estimation of $\mathcal{X}_\tau^*$ into accurate clustering of nodes. Lemma 3.3 of Qin and Rohe (2013) shows that $\mathcal{X}_\tau^*$ can be written as $\mathcal{X}_\tau^* = \mathbf{MB}$, where $\mathbf{B}$ is a K-by-K non-singular matrix. Note that the membership matrix $\mathbf{M}$ has exactly K unique rows. Hence, $\mathcal{X}_\tau^*$ also has exactly K unique rows. This implies that a K-means clustering applied on the rows of $\mathcal{X}_\tau^*$ would perfectly identify the block membership of all nodes in the network. Given the asymptotic closeness between $\mathbf{X}_\tau^*$ and $\mathcal{X}_\tau^*$ from (2.8), one might expect that the clustering output from $\mathbf{X}_\tau^*$ also approaches the clustering output from $\mathcal{X}_\tau^*$ asymptotically. As discussed above, the clustering output from $\mathcal{X}_\tau^*$ is perfect; hence, the clustering output from $\mathbf{X}_\tau^*$ is expected to approach that perfect accuracy in an asymptotic sense.

To formalize this intuition, a mathematically tractable definition of misclustering is required. In step 4 of the regularized spectral clustering algorithm, the N rows of $\mathbf{X}_\tau^*$ are subjected to a K-means clustering, which assigns each row to a cluster, and each cluster thus formed will have a centroid. Let $\mathbf{C}$ be the N-by-K matrix of cluster centroids, i.e., $\mathbf{C}(i,\cdot)$ is the centroid corresponding to the i^{th} row of $\mathbf{X}_\tau^*$. Then $\mathcal{X}_\tau^*(i,\cdot)$ is the parameter centroid corresponding to the i^{th} node, while the estimated centroid is $\mathbf{C}(i,\cdot)$. It is therefore reasonable to consider the i^{th} node to be correctly clustered if the estimated centroid is closer to the correct parameter centroid than the remaining $K-1$ incorrect parameter centroids, and it is misclustered if the estimated centroid is closer to some incorrect parameter centroid than the correct parameter centroid.

Definition The set of misclustered nodes $\mathcal{E}$ is defined as

$$\mathcal{E} = \{i : \exists\, j \neq i \; s.t. \; ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^*(i,\cdot)\mathbf{O}||_2 > ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^*(j,\cdot)\mathbf{O}||_2\}\,. \tag{2.9}$$

For some $\epsilon > 0$ and sufficiently large N, suppose Assumptions $(A1)$ and $(A2)$

hold. Then Theorem 4.4 of Qin and Rohe (2013) states that, with probability at least $1 - \epsilon$,

$$|\mathcal{E}| \leq c_1 \frac{K \log(N/\epsilon)}{\gamma^2 \lambda^2 (\delta + \tau)} \tag{2.10}$$

for some constant c_1.

Next, we extend these ideas to the Het-DCBM and the Het-RSC algorithm. For the T-type, K-block Het-DCBM from Section 2.3.2, $\mathbf{M}$ has TK unique rows, and $\mathbf{P}$ is TK-by-TK, since link probabilities are allowed to vary for each type-block combination. Note that this model is structurally equivalent to a Hom-DCBM (3.3) with TK blocks. The interpretation of sub-blocks in a T-type, K-block heterogeneous model is different from that of blocks in a TK-block homogeneous model, but both models have the same mathematical structure. Consequently, the convergence result for row-normalized eigenvectors in (2.8) can be directly applied to the heterogeneous model.

The translation of estimation accuracy to clustering accuracy, however, does not extend directly from the homogeneous version to the heterogeneous version. The upper bound in (2.10) for the homogeneous case is derived from the fact that the matrix of cluster centroids, $\mathbf{C}$, is the minimizer of the K-means objective function $||\mathbf{X}^*_\tau - \mathbf{Y}||_F$, minimization being performed over the set of all N-by-K matrices $\mathbf{Y}$ having exactly K unique rows. The Het-RSC algorithm in Section 2.4.2 runs T separate K-means procedures on the T node types, thereby using a different objective function. Therefore, (2.10) does not apply directly to heterogeneous networks. However, after considering the modified objective function being minimized in step 4 of the Het-RSC algorithm, we are able to prove the following theorem, which provides the heterogeneous version of (2.10).

Theorem 2.5.1 *Consider a T-type, K-block Het-DCBM with n_t nodes of type t, and $N = \sum_{t=1}^{T} n_t$. For nodes of type t, define the set of misclustered nodes $\mathcal{E}_t$ as*

$$\begin{aligned} \mathcal{E}_t = \{i \in \textit{type } t : \exists\, j \in \textit{type } t \;\&\; j \neq i \textit{ s.t.} \\ ||\mathbf{C}(i,\cdot) - \mathcal{X}^*_\tau(i,\cdot)\mathbf{O}||_2 > ||\mathbf{C}(i,\cdot) - \mathcal{X}^*_\tau(j,\cdot)\mathbf{O}||_2\}. \end{aligned} \tag{2.11}$$

Let $\gamma_t = \min_{i \in \textit{type } t}\{\min\{||\mathbf{X}_\tau(i,\cdot)||_2, ||\mathcal{X}_\tau(i,\cdot)||_2\}$ be the length of the shortest row of type t in $\mathcal{X}_\tau$ and $\mathbf{X}_\tau$, and λ, δ, τ be defined as before. For some $\epsilon > 0$ and sufficiently large N, suppose Assumptions (A1) and (A2) hold. Then with probability at least $1 - \epsilon$,

$$|\mathcal{E}_t| \leq c_1 \frac{K \log(N/\epsilon)}{\gamma_t^2 \lambda^2 (\delta + \tau)} \quad \textit{for } t = 1, \ldots, T, \tag{2.12}$$

where c_1 is some constant.

The proof of Theorem 1 is in the Appendix.

Remark 1 Two main assumptions of Theorem 1 are $(A1)$ and $(A2)$. Assumption $(A1)$ requires a lower bound on the smallest regularized expected degree $\delta + \tau$. This emphasizes the importance of regularization, as it allows expected node degrees to be low, as long as they are complemented by the regularizer. Assumption $(A2)$ requires that the smallest non-zero eigenvalue of $\mathcal{L}_\tau$ in magnitude does not decay to zero too fast. The number of types, T, is arbitrary but fixed. The number of blocks, K, is allowed to increase with n_t and N as long as Assumptions $(A1)$ and $(A2)$ hold true — thus its allowable rate on increase depends on the large sample behavior of the quantities δ, τ, and λ.

Theorem 1 provides a separate bound for each node type. Under Assumptions $(A1)$ and $(A2)$, for a given type t and for sufficiently large N, the quantity on the right hand side of (2.12) is $O(1/\gamma_t^2)$. Therefore as $n_t \to \infty$, the asymptotic bound on the number of misclustered nodes depends on the behavior of γ_t. When γ_t decays at a rate slower than $\sqrt{1/n_t}$, the bound in (2.12) implies that the error rate $|\mathcal{E}_t|/n_t$ goes to zero. The bound deteriorates when γ_t decays to zero faster than $\sqrt{1/n_t}$. Note that it is plausible that the bound goes to zero for certain node types, but not for others — depending upon the behavior of γ_t for different node types.

Remark 2 (Application to Het-SC under Het-SBM) The convergence of Hom-SC under Hom-SBM was first established by Rohe et al. (2011) under the assumptions of a dense network model. Their results can be extended from the homogeneous setting to the heterogeneous setting, but that would restrict the application to the dense network case. The framework of Qin and Rohe (2013) allows for sparse networks in a broader class of degree-corrected models, and although we have focussed on Het-RSC under Het-DCBM so far in this section, the results can be readily applied to Het-SC under Het-SBM as outlined below.

The main difference between the Het-SC algorithm and the Het-RSC algorithm is that the former does not have regularization or normalization steps. The Het-SBM is a special case of the Het-DCBM, when the degree parameters θ_i are equal. In this special case, the model eigenvector matrix $\mathcal{X}_\tau$ already has exactly TK distinct rows (applying Lemma 3.3 of Qin and Rohe (2013)) corresponding to the TK sub-blocks. Hence, row-normalization is not required — we can cluster the rows of $\mathbf{X}_\tau$ directly and use the result in (2.7). Further, we can set the regularizer $\tau = 0$, i.e., no regularization. Note that in doing this the advantage of regularization is lost, and the network model is required to satisfy the following restricted version of $(A1)$ and $(A2)$:

$$(A1')\ \delta > 3\log(4N/\epsilon) \quad \text{and} \quad (A2')\ \lambda \geq 8\sqrt{\frac{3K\log(4N/\epsilon)}{\delta}}.$$

Here λ is the magnitude of the smallest non-zero eigenvalue of $\mathcal{L}$ (the unregularized Laplacian) in magnitude. We obtain the following result for Het-SC under the Het-SBM.

Theorem 2.5.2 *Consider a T-type, K-block Het-SBM with n_t nodes of type t, and $N = \sum_{t=1}^{T} n_t$. Let $\mathbf{C}$ denote the matrix of cluster centroids resulting from Het-SC, and the set of misclustered nodes be defined as*

$$\begin{aligned}\mathcal{E}_t = \{i \in \textit{type } t : \exists\ j \in \textit{type } t \ \&\ j \neq i \textit{ s.t.} \\ \|\mathbf{C}(i,\cdot) - \mathcal{X}_{\tau=0}(i,\cdot)\mathbf{O}\|_2 > \|\mathbf{C}(i,\cdot) - \mathcal{X}_{\tau=0}(j,\cdot)\mathbf{O}\|_2\}.\end{aligned} \quad (2.13)$$

Let λ and δ be defined as before, and $\tau = 0$. For some $\epsilon > 0$ and sufficiently large N, suppose Assumptions $(A1')$ and $(A2')$ hold. Then with probability at least $1-\epsilon$,

$$|\mathcal{E}_t| \leq c_1 \frac{K\log(N/\epsilon)}{\lambda^2(\delta+\tau)} \quad \textit{for } t = 1, \ldots, T, \quad (2.14)$$

where c_1 is some constant.

The proof for Theorem 2.5.2 is essentially similar to that for Theorem 2.5.1, the only difference being the use of the eigenvector matrix $\mathbf{X}_{\tau=0}$ instead of its normalized version $\mathcal{X}^*_{\tau=0}$, and hence we skip the proof of Theorem 2.5.2 to avoid repetition.

2.6 Simulation Results

This section reports three simulation studies comparing the finite-sample performance of the homogeneous clustering algorithms with their heterogeneous counterparts in bi-type heterogeneous networks, i.e., $T = 2$. We study both Het-SBM and Het-DCBM in our simulation studies. Although Het-SBM is a special case of Het-DCBM, it is an important special case from a methodological perspective. Regularized spectral clustering adds two extra steps to standard spectral clustering - regularization (step 1) and row-normalization (step 3). The former aims to deal with sparsity (nodes with low expected degrees), while the latter aims to deal with non-uniformity in expected node degrees. In our simulations, both these features stem from degree parameters in Het-DCBM. For Het-SBM, the uniformity of expected node degrees makes both regularization and normalization unnecessary. Therefore, we study the performance of Het-SC vs Hom-SC under networks generated from Het-SBM, and that of Het-RSC vs Hom-RSC in networks generated from Het-DCBM.

The class of Het-SBM models used for these simulations is $\mathcal{B}(K; s_1, s_2, p_1, r_1, p_2, r_2, p_3, r_3)$ where K is the number of blocks, and s_1 and s_2 are the number of type 1 and type 2 nodes per block, respectively. The probability matrix is given by $\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix}$, where

$$\begin{aligned} \mathbf{P}_{11} &= p_1 \mathbf{1}_K \mathbf{1}_K' + r_1 \mathbf{I}_K, \\ \mathbf{P}_{22} &= p_2 \mathbf{1}_K \mathbf{1}_K' + r_2 \mathbf{I}_K, \\ \mathbf{P}_{12} = \mathbf{P}_{21} &= p_3 \mathbf{1}_K \mathbf{1}_K' + r_3 \mathbf{I}_K. \end{aligned}$$

Here $\mathbf{1}_K$ is a K-vector of 1's, and $\mathbf{I}_K$ is the K-by-K identity matrix. Thus, in the type 1-type 1 (type 2-type 2) homogeneous network, $p_1 (p_2)$ represents the inter-block link probability while $p_1 + r_1$ $(p_2 + r_2)$ is the intra-block link probability. The strength of homophily in the homogeneous networks is therefore determined by r_1 and r_2. For type 1-type 2 links, p_3 represents the inter-block, inter-type link probability and r_3 represents the strength of inter-type homophily. For Het-DCBM, we use the same values of $\mathbf{P}, K, s_1$, and s_2. The degree parameters θ_i are generated from the power law distribution

$$f(x) = \frac{\beta - 1}{x_{\min}} \left(\frac{x}{x_{\min}} \right)^{-\beta}$$

with $x_{\min} = 1$ and shape parameter $\beta = 3$, and then scaled down so that the average $\bar{\theta}$ equals 1. For a given parameter combination, the Het-DCBM can therefore be interpreted as a ‘noisy’ version of the Het-SBM, or conversely the Het-SBM can be interpreted as an ‘averaged’ version of the Het-DCBM. For illustration, the model used in the example in Section 2.3.2 had $K = 3$, $s_1 = s_2 = 5$, $p_1 = 0.25$, $r_1 = 0.50$, $p_2 = r_2 = p_3 = 0$, and $r_3 = 0.90$.

Our main objective in these simulations is to study how the improved accuracy of heterogeneous clustering over homogeneous clustering depends on r_3 for a fixed value of p_3. Ceteris paribus, higher values of r_3 will make the type 1-type 2 links more strongly homophilic, and therefore make heterogeneous community detection easier. Hence, we expect Het-SC and Het-RSC to be increasingly accurate with increasing r_3, while Hom-SC and Hom-RSC are not affected by r_3. However, the actual improvement of heterogeneous clustering over homogeneous clustering will depend on other parameters as well, particularly p_1, r_1, p_2 and r_2, which determine the strength of homophily (and hence the ease of community detection) for the homogeneous networks.

To capture these dynamics, we fix $K = 3, s_1 = 100, s_2 = 50$, and $p_3 = 0.25$. Thus, we study networks having a total of $N = 450$ nodes, of which 300 are of type 1 and 150 are of type 2. The parameters $p_1, p_2, r_1,$ and r_2 are set to various combinations, to generate various kinds of homogeneous networks. For each such combination, r_3 is increased in a fixed grid, from 0.10 to 0.50 in increments of 0.05. For each combination of parameters, error rates are estimated by averaging across 100 networks from Het-SBM and Het-DCBM.

It is important to note how clustering performance is measured in these simulations. Definitions (2.9), (2.11), and (2.13) introduced model-based quantification of misclustered nodes for the purpose of mathematical tractability, and the bounds (2.10), (2.12), and (2.14) were established under these definitions. However, these definitions require complete knowledge of the underlying model generating the network. For calculation of misclustering error in a real network, the true membership (ground truth) might be known (providing information about $\mathbf{M}$), but the other model parameters will generally not be known, and hence these definitions can not be evaluated for real networks. Accordingly, the error rate used in these simulations is not the model-based quantity from Section 2.5, but rather it is the usual data-based error rate, defined below as the proportion of nodes that got assigned to wrong clusters.

For a K-block network model, the *true* membership ($\mathbf{M}$) provides a K-partition, say $\mathfrak{P}$, of the nodes. Suppose there are two competing clustering algorithms which also provide K-partitions, say $\mathfrak{P}_1$ and $\mathfrak{P}_2$ respectively. For each algorithm, consider the K-by-K overlap table $\mathbf{T}_i$ such that $\mathbf{T}_i(k,l)$ is the number of nodes that have been assigned to the k^{th} block according to $\mathfrak{P}$ and to the l^{th} block according to $\mathfrak{P}_i$. Then $\sum_k \mathbf{T}_i(k,k)$ is the total number of correctly clustered nodes according to $\mathfrak{P}_i$. However, due to potential identifiability issues with cluster labels, $\mathbf{T}_i$ should not be used directly to compare the accuracy of $\mathfrak{P}_1$ and $\mathfrak{P}_2$. For example, suppose $\mathfrak{P}_1$ gives the same partition as $\mathfrak{P}$, but with different cluster labels, while $\mathfrak{P}_2$ is just a random partition. Then $\sum_k \mathbf{T}_1(k,k) = 0$ and hence $\sum_k \mathbf{T}_2(k,k) > \sum_k \mathbf{T}_1(k,k)$, but $\mathfrak{P}_1$ is clearly more accurate than $\mathfrak{P}_2$. This issue can be resolved by permuting the columns of $\mathbf{T}_i$ to maximize $\sum_k \mathbf{T}_i(k,k)$. This is equivalent to relabeling the clusters of $\mathfrak{P}_i$, and makes no change to the K-partition of the nodes. Now it is reasonable to compare $\mathfrak{P}_1$ and $\mathfrak{P}_2$ on the basis of permuted $\mathbf{T}_1$ and $\mathbf{T}_2$. The error rate is defined as the sum of off-diagonal elements of $\mathbf{T}_i$, divided by the total number of nodes.

Thus, these simulations are not aimed at studying the finite-sample behavior of the quantities involved in the asymptotic theory, rather they are aimed at studying the relative accuracy of homogeneous and heterogeneous clustering methods in finite networks generated from Het-SBM and Het-DCBM.

2.6.1 Simulation 1

In this simulation we study homophilic networks where both types have similar inter-block and intra-block link probability. We use $p_1 = p_2 = 0.25$, and use a single parameter, $r_1 = r_2 = r$, say, to govern the strength of homophily in the homogeneous networks. Two values of $r = 0.10, 0.15$ are used, to construct homogeneous networks of two different strengths. The error rates from Het-SBM and Het-DCBM are plotted in Figure 2.2(a) and Figure 2.3(a) respectively.

It is observed in both models that homogeneous error rates for type 1 nodes are substantially lower than type 2, implying the effect of sample size or block size on error rates. Homogeneous error rates also go down quite remarkably for both types as r is increased from 0.10 to 0.15, since the homogeneous network becomes more strongly homophilic, rendering community detection easier. This phenomenon is more prominent in Het-SBM (Figure 2.2(a)) than Het-DCBM (Figure 2.3(a)). However, the most striking observation in Figure 2.2(a) and Figure 2.3(a) is the improved accuracy of heterogeneous clustering over homogeneous clustering, for both types and both models. This comparative advantage increases with increasing r_3, but it is significant even for smaller values of r_3.

2.6.2 Simulation 2

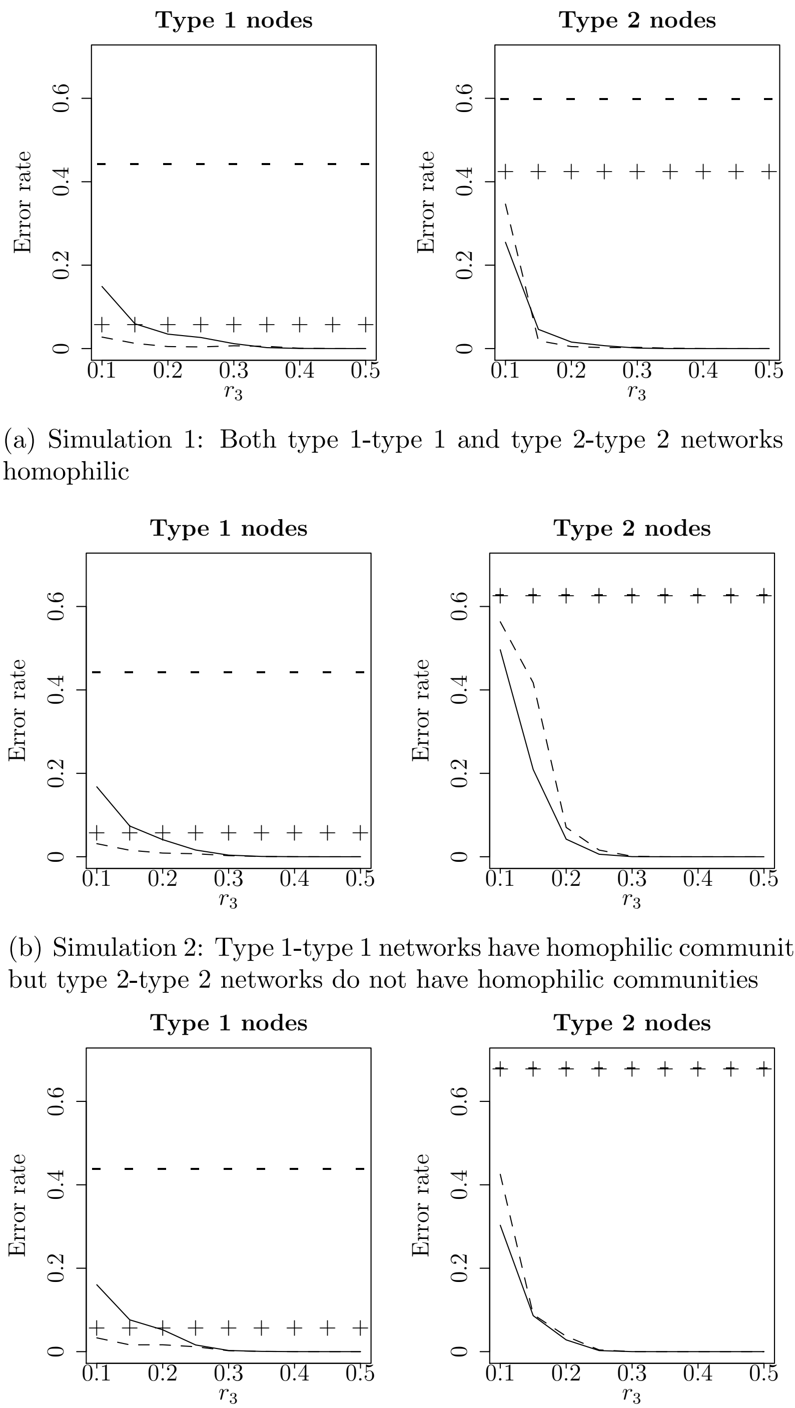
A plausible scenario in heterogeneous networks is when the type 1-type 1 homogeneous network is homophilic but the type 2-type 2 network does not have homophilic community structure. To model this, we use $p_1 = p_2 = 0.25$ as before, but set the type 2 homophily parameter $r_2 = 0$ while the type 1 homophily parameter r_1 is increased from 0.1 to 0.15.

Figure 2.2(b) and Figure 2.3(b) show that the heterogeneous methods are much more accurate for both node types. The improved accuracy over the homogeneous method is particularly remarkable for type 2 nodes, because the homogeneous type 2-type 2 network does not have homophilic communities, and it would be quite difficult to assign communities to nodes on the basis of homogeneous information only. For example, consider a high school social network where students (type 1) form homophilic communities based on grades, but teachers (type 2) do not show homophily, rather they interact uniformly with other teachers. However, the heterogeneous student-teacher interaction is expected to be homophilic, as a student from a particular grade is expected to have more interaction with a teacher from the same grade, compared to a teacher from a different grade. In such a scenario, using a heterogeneous student-teacher network will most likely perform better community detection for both students and teachers, compared to clustering the homogeneous student-student network or the homogeneous teacher-teacher network, even though teachers do not interact in homophilic fashion.

2.6.3 Simulation 3

Another plausible situation is that type 1-type 1 interactions are homophilic but there is no type 2-type 2 interaction at all. We use $p_1 = 0.25$ and increase r_1 from 0.1 to 0.15 as before, but set $p_2 = r_2 = 0$, so that there are no links between type 2 nodes. A motivation for this situation is the notional Facebook user-event heterogeneous network described in the introduction. While users (type 1) form a homophilic friendship network with universities as communities, there is no natural interaction between two events (type 2), implying a blank type 2-type 2 network. However, there is expected to be strong homophily in user-event interactions, and hence it is quite likely that the heterogeneous method will deliver a superior performance than the homogeneous method.

Figure 2.2(c) and Figure 2.3(c) show that the heterogeneous method is indeed significantly superior to the homogeneous method, for both types. In this case, it is theoretically impossible to implement homogeneous spectral clustering on the type 2-type 2 network, as the Laplacian for this network is a zero matrix, while the heterogeneous method delivers quite accurate clustering for type 2 nodes. For the sake of comparison, we have used a flat homogeneous error rate of 2/3 (random allocation with $K = 3$ clusters) for type 2 nodes.

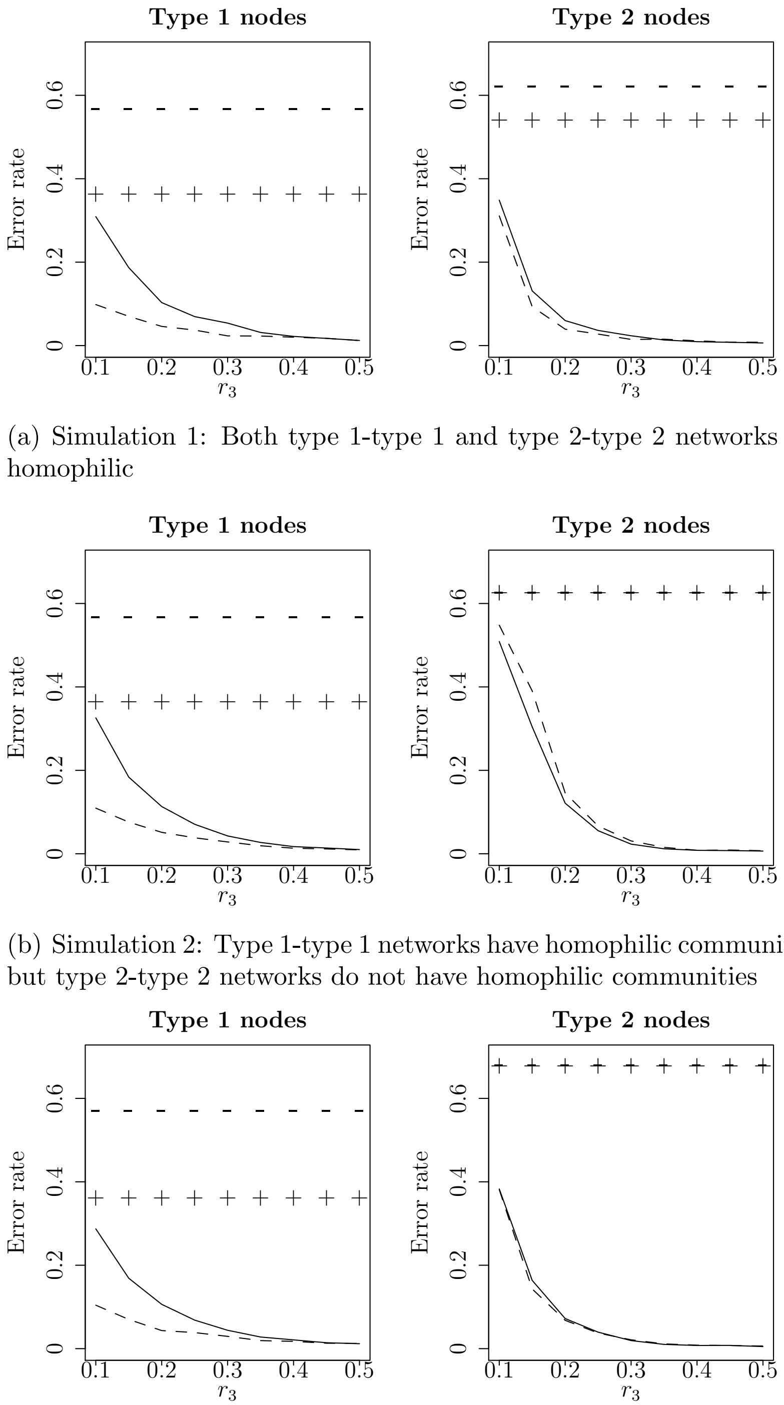


(a) Simulation 1: Both type 1-type 1 and type 2-type 2 networks are homophilic

(b) Simulation 2: Type 1-type 1 networks have homophilic communities but type 2-type 2 networks do not have homophilic communities

(c) Simulation 3: Homophilic type 1-type 1 networks and no type 2-type 2 links

Figure 2.2: For simulation 1, Hom-SC errors are represented as '-' for $r_1 = r_2 = 0.1$ and '+' for $r_1 = r_2 = 0.15$, while Het-SC errors are represented by solid lines for $r_1 = r_2 = 0.1$ and dashed lines for $r_1 = r_2 = 0.15$. For simulations 2 and 3, Hom-SC errors are represented as '-' for $r_1 = 0.1$ and '+' for $r_1 = 0.15$, while Het-SC errors are represented by solid lines for $r_1 = 0.1$, and dashed lines for $r_1 = 0.15$.



(a) Simulation 1: Both type 1-type 1 and type 2-type 2 networks are homophilic

(b) Simulation 2: Type 1-type 1 networks have homophilic communities but type 2-type 2 networks do not have homophilic communities

(c) Simulation 3: Homophilic type 1-type 1 networks and no type 2-type 2 links

Figure 2.3: For simulation 1, Hom-RSC errors are represented as '-' for $r_1 = r_2 = 0.1$ and '+' for $r_1 = r_2 = 0.15$, while Het-RSC errors are represented by solid lines for $r_1 = r_2 = 0.1$ and dashed lines for $r_1 = r_2 = 0.15$. For simulations 2 and 3, $r_2 = 0$, and Hom-RSC errors are represented as '-' for $r_1 = 0.1$ and '+' for $r_1 = 0.15$, while Het-RSC errors are represented by solid lines for $r_1 = 0.1$, and dashed lines for $r_1 = 0.15$.

2.7 DBLP Four-Area Dataset Example

DBLP (Digital Bibliography & Library Project) is the authoritative computer science bibliography website, listing over two million articles. Gao et al. (2009) and Ji et al. (2010) extracted a connected subset of the DBLP data, containing bibliographical records from four research areas related to data mining, namely *database*, *data mining*, *information retrieval* and *artificial intelligence*. The clustering problem of interest is to identify research area for authors. The original four-area dataset consists of 14,376 papers written by 14,475 authors, and presented at 20 conferences. However, the *ground truth* (true research area) is available for 4,057 authors, who account for 14,328 of these papers, covering all 20 conferences. Since error rates can be calculated only for labeled authors, our data analysis is based on this labeled subset of the data.

In the simulation studies of Section 2.6, we implemented Het-SC and Hom-SC on Het-SBM, and Het-RSC and Hom-RSC on Het-DCBM, backed by theoretical justification. However, in real-world applications, we have to choose between standard and regularized spectral clustering, for both homogeneous and heterogeneous networks, on the basis of empirical features. In general, we expect regularization to work better if the network is sparse. Two distinguishing properties that are found in many large real-world sparse networks (Girvan and Newman (2002)) are (i) a large number of nodes with low degrees, and (ii) power law behavior of degrees. We plot the histogram and log empirical tail distribution $\log_{10}(1 - \hat{F}(x))$ of node degrees in Figure 2.4 to investigate these properties. A heavily right-skewed histogram will indicate property (i) and a roughly linear plot of $\log_{10}(1 - \hat{F}(x))$ will indicate property (ii). Accordingly, in the following analysis we choose regularization if the plots indicate sparsity.

2.7.1 Homogeneous author collaboration network

For homogeneous clustering, the natural network is the co-authorship network, where authors are nodes, and two authors are linked if they have collaborated to write a paper. Authors belonging to the same research area are more likely to collaborate, so the network has homophilic structure with research areas as communities. This gives a homogeneous network with 4,057 connected nodes and 3,528 links. Figure 2.4 (left column) shows a heavily right-skewed histogram and a roughly linear log empirical tail distribution plot, indicating that we should prefer Hom-RSC over Hom-SC for clustering this network.

However, 1466 of the 4057 authors have no edges in the author-author network and hence they have to be discarded, as disconnected nodes cannot be clustered either by Hom-SC or by Hom-RSC. We implement the Hom-RSC algorithm from Section 2.4 on the remaining 2,591 nodes with $K = 4$ clusters. It turns out that 482 rows of the eigenvector matrix $\mathbf{X}$ are null rows — these rows can not be row-normalized and hence the corresponding nodes cannot be clustered. After discarding these nodes as well, we perform clustering on the remaining 2109 author nodes. The algorithm misclusters 1274 (60.41%) of these nodes. If we randomly assign the discarded nodes to the 4 clusters, the weighted average error rate for all 4057 nodes is 67.41%. This number is the weighted average of clustering error (60.41%) and random assignment error (75%). We also implement the Hom-SC algorithm — the error rate is 69.74% for the 2,591 connected authors and the weighted error rate for all authors is 71.64%. Note that Hom-SC does not have a problem with clustering null rows in the eigenvector matrix. Thus, while Hom-RSC does perform better than Hom-SC, both homogeneous algorithms have accuracy comparable to random assignment to clusters.

2.7.2 Heterogenous author-paper-conference network

Note that the DBLP system consists of several *types* of entities, namely *authors*, *papers*, *conferences*. A heterogeneous network representation (APC network) of the DBLP system can thus be constructed with three types of nodes: authors, papers, and conferences, and two types of links: author-paper (author writes paper) links and paper-conference (paper presented at conference) links. Authors are more likely to write papers in their research area, and papers are more likely to be presented at a conference belonging to the same research area, indicating homophilic community structure. This gives a network with 18,405 nodes (4,057 authors, 14,328 papers, 20 conferences) and 33,973 links (19,645 author-paper links and 14,328 paper-conference links). All authors are now connected. The middle column of Figure 2.4 shows a heavily right-skewed histogram and a roughly linear log empirical tail distribution plot for author node degrees, indicating that we should prefer Het-RSC over Het-SC for clustering this network.

We implement the Het-RSC algorithm from Section 2.4 on this network with $T = 3$ and $K = 4$. The error rate for authors is 7.30%. We also implement Het-SC which gives an error rate of 23.10% for the authors. Thus, Het-RSC is quite accurate in identifying research area for authors from the heterogeneous network. Even Het-SC performs relatively well, although Het-RSC is more accurate than

Het-SC as expected for a sparse network. In contrast, the homogeneous algorithms have accuracy similar to random allocation, which implies that the homogeneous co-authorship network is not very informative towards identification of authors' research area.

2.7.3 Heterogeneous author-conference network

The problem of interest in the four-area DBLP dataset is assigning authors to research communities, which is a homogeneous problem relating only to author nodes. However, the DBLP system itself is heterogeneous, and this heterogeneous information can be useful towards solving this homogeneous problem. In section 2.7.2, we used data from the heterogeneous DBLP system to add two additional types of nodes (papers and conferences) to construct a heterogeneous network. This is the 'default' way to construct the heterogeneous network, using all the data at our disposal, and this approach gives us a much better solution to the problem than the homogeneous approach in Section 2.7.1.

However, suppose we instead consider a heterogeneous sub-system, and add only conference nodes, forming a smaller heterogeneous network with two types of nodes (authors and conferences) and only one type of link, author-conference (author presented at the conference). Authors from a research area are more likely to present at a conference related to the same area, indicating a homophilic community structure. This gives a network with 4,077 nodes (4,057 authors and 20 conferences) and 9,205 author-conference links. All authors are connected. The right column of Figure 2.4 shows a histogram that is right-skewed but not as heavily right-skewed as the two earlier networks. The log empirical tail distribution plot is also less linear than the other two networks. The node degrees vary between 1 and 14, which is a much tighter range than the degree range in the homogeneous author network or the heterogeneous APC network. Thus the network features do indicate sparsity, but less so than the two previous DBLP networks.

Implementing the Het-RSC algorithm on this bi-type network, we obtain an error rate of 7.44%, which is comparable to the error rate of Het-RSC in the APC network. The Het-SC algorithm gives an error rate of 8.85%, which is better than Het-SC in APC network. Both error rates are significant improvement over the homogeneous approach. Note that such improvement is achieved with only 20 additional nodes and therefore at a computational cost comparable to the homogeneous approach, while the APC network requires the addition of 14,348 nodes and therefore has greater computational cost.

In general, often the community detection problem of interest itself is homogeneous, in the sense that it is defined with respect to only one type of agents, while the underlying system is heterogeneous. The user has the flexibility to choose from several heterogeneous sub-systems of the data to create a heterogeneous network. For example in the DBLP system, the user can choose the entire author-paper-conference system, or the author-conference subsystem, and so on. Consequently, the user might be interested in using an *optimal sub-system* that delivers the best community detection for the problem. One interesting avenue of future work is to lay down explicit criteria for selecting the optimal sub-network, akin to the analogous problem of variable selection in a machine learning framework.

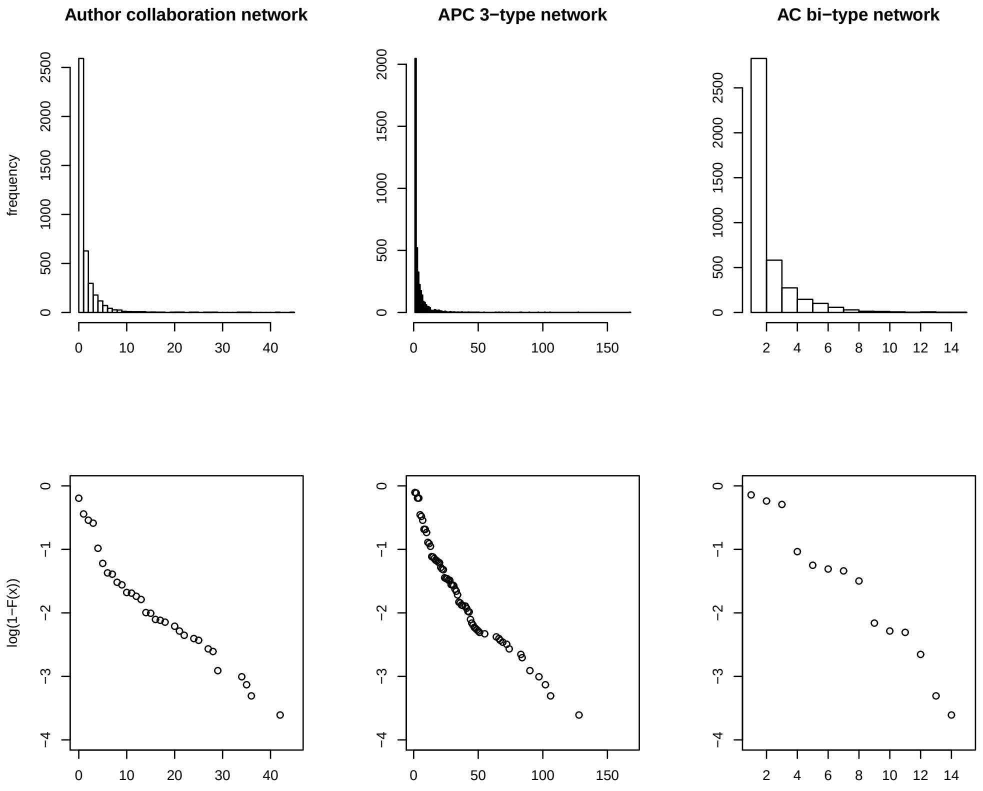


Figure 2.4: DBLP author degree distribution of homogeneous author collaboration network (left column), heterogeneous author-paper-conference network (middle column), and heterogeneous author-conference network (right column). Histograms (top row) of author node degrees have high frequency of low degrees, indicating that the author nodes are sparsely connected. The bottom row shows that the log empirical tail distributions $\log_{10}(1 - \hat{F}(x))$ are roughly linear, suggesting power-law behavior of author node degrees.

2.8 Discussion

This paper introduces heterogeneous networks to the statistics literature, and extends the existing statistical framework of community detection in homogeneous networks to heterogeneous networks. We formulate heterogeneous versions of standard spectral clustering and regularized spectral clustering algorithms. The proposed algorithms have theoretical accuracy under heterogeneous versions of the SBM and the DCBM, respectively. Our simulations demonstrate that even though homogeneous and heterogeneous methods have similar order of theoretical accuracy in large samples, the heterogeneous methods provide significantly better clustering results in finite-sample networks generated from several interesting model settings. This comparative advantage seems to imply that the superiority of heterogeneous clustering over homogeneous clustering should be theoretically demonstrable, however we leave that theoretical exercise as future work. The practical usefulness of the heterogeneous procedure is also demonstrated by the DBLP four-area dataset example, where the heterogeneous method delivers a far better clustering performance compared to the homogeneous method.

2.9 Proof of Theorem 1

We prove the theorem for $T = 2$, i.e., bi-type heterogeneous networks. The proof easily generalizes to higher values of T. Consider a K-block, bi-type Het-DCBM with n_1 nodes of type 1 and n_2 nodes of type 2, and let $\tau \geq 0$ be the regularizer. Let $N = n_1 + n_2$ and let $\mathbf{X}_\tau, \mathcal{X}_\tau, \mathbf{X}^*_\tau,$ and $\mathcal{X}^*_\tau$ be N-by-$2K$ matrices defined as per Sections 2.4 and 2.5.

Partition $\mathbf{X}^*_\tau$ as $\mathbf{X}^*_\tau = \begin{pmatrix} \mathbf{X}^{*(1)}_\tau \\ \mathbf{X}^{*(2)}_\tau \end{pmatrix}$, where $\mathbf{X}^{*(1)}_\tau$ is n_1-by-$2K$ and $\mathbf{X}^{*(2)}_\tau$ is n_2-by-$2K$. Then cluster centroids are given by:

$$\mathbf{C}_t = \arg\min_{\mathbf{Y}_t \in \mathcal{Y}_t} ||\mathbf{X}^{*(t)}_\tau - \mathbf{Y}_t||_F^2 \;\text{ for }\; t = 1, 2, \tag{2.15}$$

where $\mathcal{Y}_t = \{\mathbf{Y}_t \in \mathcal{R}^{n_t \times 2K} : \mathbf{Y}_t \text{ has } K \text{ unique rows}\}$, for $t = 1, 2$.

Note that, for the bi-type Het-DCBM, $\mathbf{M}$ has the form $\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{22} \end{pmatrix}$, where $\mathbf{M}_{11}$ is n_1-by-K with exactly K distinct rows, and $\mathbf{M}_{22}$ is n_2-by-K with exactly K distinct rows. By Lemma 3.3 (2) of Qin and Rohe (2013), $\mathcal{X}^*_\tau$ can be expressed as $\mathcal{X}^*_\tau = \mathbf{MB}$ under the general DCBM, and hence also under the Het-DCBM, where $\mathbf{B}$ is a non-singular matrix of dimension $2K$-by-$2K$. Partition $\mathbf{B}$ into four K-by-K matrices as $\mathbf{B} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix}$. Then,

$$\mathcal{X}_\tau^* = \mathbf{MB} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{11}\mathbf{B}_{11} & \mathbf{M}_{11}\mathbf{B}_{12} \\ \mathbf{M}_{22}\mathbf{B}_{21} & \mathbf{M}_{22}\mathbf{B}_{22} \end{pmatrix} = \begin{pmatrix} \mathcal{X}_\tau^{*(1)} \\ \mathcal{X}_\tau^{*(2)} \end{pmatrix},$$

where $\mathcal{X}_\tau^{*(1)}$ is n_1-by-$2K$ and $\mathcal{X}_\tau^{*(2)}$ is n_2-by-$2K$. Since $\mathbf{M}_{11}$ and $\mathbf{M}_{22}$ have exactly K unique rows, and $\mathbf{B}$ is non-singular, both $\mathcal{X}_\tau^{*(1)}$ and $\mathcal{X}_\tau^{*(2)}$ have K distinct rows, that is, $\mathcal{X}_\tau^{*(1)} \in \mathcal{Y}_1$ and $\mathcal{X}_\tau^{*(2)} \in \mathcal{Y}_2$. This also implies that $\mathcal{X}_\tau^{*(t)}\mathbf{O} \in \mathcal{Y}_t$ for $t = 1, 2$, where $\mathbf{O}$ is an orthonormal rotation.

Without loss of generality we focus on $t = 1$. From the definition of $\mathbf{C}_1$ and the fact that $\mathcal{X}_\tau^{*(1)}\mathbf{O} \in \mathcal{Y}_1$, $||\mathbf{X}_\tau^{*(1)} - \mathbf{C}_1||_F \leq ||\mathbf{X}_\tau^{*(1)} - \mathcal{X}_\tau^{*(1)}\mathbf{O}||_F$. So,

$$||\mathbf{C}_1 - \mathcal{X}_\tau^{*(1)}\mathbf{O}||_F \leq ||\mathbf{C}_1 - \mathbf{X}_\tau^{*(1)}||_F + ||\mathbf{X}_\tau^{*(1)} - \mathcal{X}_\tau^{*(1)}\mathbf{O}||_F \leq 2||\mathbf{X}_\tau^{*(1)} - \mathcal{X}_\tau^{*(1)}\mathbf{O}||_F.$$

Recall that $\mathcal{E}_1$ is defined as

$$\mathcal{E}_1 = \{i \in \text{type } 1 : \exists\ j \in \text{type } 1\ \&\ j \neq i\ s.t. \\ ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^*(i,\cdot)\mathbf{O}||_2 > ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^*(j,\cdot)\mathbf{O}||_2\}.$$

Note that for two type 1 nodes $i \neq j$, either $M(i,\cdot) = M(j,\cdot)$ when they belong to the same block, or $M(i,\cdot)'M(j,\cdot) = 0$ when they belong to different blocks. Since $\mathcal{X}_\tau^* = \mathbf{MB}$, $\mathcal{X}_\tau^*$ is row-normalized, and $\mathbf{O}$ is orthonormal, this implies that for two type 1 nodes $i \neq j$, either

$$\mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O} = \mathcal{X}_\tau^{*(1)}(j,\cdot)\mathbf{O} \Rightarrow ||\mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O} - \mathcal{X}_\tau^{*(1)}(j,\cdot)\mathbf{O}||_2 = 0$$

or

$$(\mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O})'(\mathcal{X}_\tau^{*(1)}(j,\cdot)\mathbf{O}) = 0 \Rightarrow ||\mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O} - \mathcal{X}_\tau^{*(1)}(j,\cdot)\mathbf{O}||_2 = \sqrt{2}.$$

This leads to the observation that

$$||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O}||_2 < \frac{1}{\sqrt{2}} \Rightarrow \\ ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O}||_2 \leq ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^{*(1)}(j,\cdot)\mathbf{O}||_2,\ \forall\ j \neq i.$$

which means $||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O}||_2 < 1/\sqrt{2}$ is a sufficient condition for node i to be correctly clustered. Define $\mathcal{E}_1'$ to be the set of nodes that do not satisfy this sufficient condition, i.e.,

$$\mathcal{E}_1' = \{i \in \text{type } 1 : ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O}||_2 \geq 1/\sqrt{2}\}.$$

Then,

$$\begin{aligned} |\mathcal{E}_1| \leq |\mathcal{E}_1'| = \sum_{i\in\mathcal{E}_1'} 1 \leq 2\sum_{i\in\mathcal{E}_1'} ||\mathbf{C}(i,\cdot) - \mathcal{X}_\tau^{*(1)}(i,\cdot)\mathbf{O}||_2^2 \leq 2||\mathbf{C}_1 - \mathcal{X}_\tau^{*(1)}\mathbf{O}||_F^2 \\ \leq 8||\mathbf{X}_\tau^{*(1)} - \mathcal{X}_\tau^{*(1)}\mathbf{O}||_F^2. \end{aligned}$$

From (2.7), we have

$$||\mathbf{X}_\tau - \mathcal{X}_\tau\mathbf{O}||_F \leq c_0\frac{1}{\lambda}\sqrt{\frac{K\log(4N/\epsilon)}{\delta+\tau}}$$

under Assumptions $(A1)$ and $(A2)$. Note that for any i,

$$||\mathbf{X}_\tau^*(i,) - \mathcal{X}_\tau^*(i,)\mathbf{O}||_2 \leq \frac{||\mathbf{X}_\tau(i,) - \mathcal{X}_\tau(i,)\mathbf{O}||_2}{\min\{||\mathbf{X}_\tau(i,)||_2, ||\mathcal{X}_\tau(i,)||_2\}}.$$

Therefore, from the definition of γ_1,

$$8||\mathbf{X}_\tau^{*(1)} - \mathcal{X}_\tau^{*(1)}\mathbf{O}||_F^2 \leq \frac{8||\mathbf{X}_\tau^{(1)} - \mathcal{X}_\tau^{(1)}\mathbf{O}||_F^2}{\gamma_1^2} \leq \frac{8||\mathbf{X}_\tau - \mathcal{X}_\tau\mathbf{O}||_F^2}{\gamma_1^2} \leq 8c_0^2\frac{K\log(4N/\epsilon)}{\lambda^2\gamma_1^2(\delta+\tau)}.$$

This completes the proof for $T = 2$.

CHAPTER 3

A BLOCKMODEL FOR NODE POPULARITY IN NETWORKS WITH COMMUNITY STRUCTURE

3.1 Introduction

Networks are ubiquitous in today's world, as a wide range of systems, such as social interpersonal systems (Milgram, 1967), power grids (Watts and Strogatz, 1998), the World Wide Web (Huberman and Adamic, 1999), and protein interaction systems (Gavin et al., 2002), to name a few, can be represented as networks. Accordingly, there has been a lot of recent emphasis in the statistics literature towards developing statistical methodology for analyzing network data. Broad overviews of network data analysis can be found in Kolaczyk (2009), Goldenberg et al. (2010), and Newman (2010).

A well-known feature of many empirical networks is community structure. Nodes in a network are often found to belong to groups or communities that exhibit similar behavior. A classical random graph model for networks with community structure is the stochastic blockmodel (SBM, hereafter) that was introduced by Lorrain and White (1971), Holland et al. (1983) and Fienberg et al. (1985). Under the SBM, two nodes belonging to the same community display community structure by behaving identically, in a stochastic sense. In particular, any two nodes from the same community have the same degree distribution and the same expected degree. This implies unrealistic structural constraints since empirical networks exhibit wide degree distributions even for nodes belonging to the same community. The degree-corrected blockmodel (DCBM, hereafter) proposed by Karrer and Newman (2011) 'corrects' this anomaly by assigning a degree parameter to each node, thereby allowing nodes in the same community to have different degree distributions.

A network feature that is closely associated with community structure is the popularity of nodes across communities, defined as the number of edges between a specific node and a specific community. Just like node degrees vary widely between nodes in the same community in empirical networks, node popularities also exhibit various kinds of patterns between nodes in the same community. In particular, for two nodes i and j in the same community, i may be more popular than j in community A, while j is more popular than i in community B. However, both the DCBM and the SBM have implicit structural constraints that lead to unrealistic restrictions on node popularity, in the same manner as the SBM has structural constraints that lead to unrealistic restrictions on node degree. The SBM restricts node popularities to behave identically for nodes in the same community. Under the DCBM, each node has a degree parameter that uniformly inflates or deflates the node's popularity in all communities, which has the unrealistic implication that for two nodes i and j in the same community, if i is more popular than j in one community, i must also be more popular than j uniformly across all communities.

To see how these restrictions under the DCBM might be unrealistic, we look at two empirical illustrations — one from the political blogs network (Adamic and Glance, 2005) in Table 3.1 and one from the Twitter network of British members of Parliament or MPs (Greene and Cunningham, 2013) in Table 3.2. We consider the DCBM fitted to both networks using the extreme points (EP) algorithm of Le et al. (2015) with regularization. We cover the datasets and analysis methodology in Section 3.6 in a comprehensive manner, the current reference is only for a quick illustration.

	Observed (Fitted by DCBM)			
Name	Community	Liberal Pop	Conservative Pop	Degree
andrewsullivan.com	conservative	58 (10)	85 (133)	143 (142)
blogsforbush.com	conservative	5 (21)	296 (278)	301 (299)
democraticunderground.com	liberal	59 (85)	34 (7)	93 (93)
liberaloasis.com	liberal	169 (157)	2 (13)	171 (170)

Table 3.1: Illustrative nodes for political blogs, with popularities fit by DCBM inside parantheses.

	Observed (Fitted by DCBM)			
Name	Community	Conservative Popularity	Labour Popularity	Degree
Zac Goldsmith	conservative	46 (62)	25 (8)	71 (70)
Matt Hancock	conservative	68 (62)	3(8)	71(70)
Seema Malhotra	labour	0 (4)	88 (84)	88 (88)
Ian Austin	labour	11 (3)	76 (83)	87 (87)

Table 3.2: Illustrative nodes for British MPs. Identities were looked up using tweeterid.com.

In Table 3.1, the degree of the node blogsforbush.com is more than twice the degree of the node andrewsullivan.com, however the latter node is almost 12 times more popular than the former among liberals, although both nodes belong to the conservative community. This is not a chance occurrence, since Andrew Sullivan is a conservative blogger with well-known liberal connections — in fact he was ranked by Forbes magazine at No. 19 on a list of “The 25 Most Influential Liberals in the U.S. Media” (Varadarajan et al., 2009), to which he wrote a swift rebuttal in Sullivan (2009). However, the DCBM enforces node popularity to be uniformly proportional to node degree, and hence the DCBM fit grossly underestimates the popularity of andrewsullivan.com among liberals and grossly overestimates the popularity of blogsforbush.com among liberals. A similar phenomenon holds for liberal blogs democraticunderground.com and liberaloasis.com, where the DCBM grossly underestimates the former node’s popularity in the conservative community, while grossly overestimating the latter node’s popularity in the conservative community. Note that the DCBM correctly detects the communities of all four blogs, and accurately fits the total degree of all four nodes, by dint of node-specific degree parameters. However the model fitting for node popularity is quite inaccurate.

Similarly for the British MP network in Table 3.2, conservative MPs Zac Goldsmith and Matt Hancock have equal degrees in the network, and hence according to DCBM, they should have identical popularity in either community. However, Zac Goldsmith is a well-known environmentalist (d’Ancona, 2015) and is much more popular among the Labour party MPs than Matt Hancock. The DCBM severely underestimates Zac Goldmith’s popularity in the Labour party while severely overestimating Matt Hancock’s popularity in the Labour party. A similar contrast holds for Labour MPs Seema Malhotra and Ian Austin.

Thus the DCBM fails to model node popularities in a flexible and realistic way. In this chapter we introduce a new random graph model, called the popularity-adjusted blockmodel (PABM, hereafter) for modeling node popularity in networks with community structure. In contrast to the SBM and the DCBM, the PABM allows flexible and realistic modeling of node popularity. We develop methodology for community detection and parameter estimation under the PABM, and revisit the above tables in Section 3.6 to demonstrate the improvement achieved through this new methodology.

The rest of the chapter is organized as follows. In Section 3.2 we formulate the PABM and compare it to the SBM and the DCBM. In Section 3.3 we derive likelihood modularity for community detection under the PABM. Section 3.4 demonstrates the consistency of likelihood modularity. We compare the performance of the PABM and the DCBM in a simulation study in Section 3.5, and in analyzing the political blogs network and the British MP network in Section 3.6. The chapter concludes with the discussion in Section 3.7 and proofs of theoretical results.

3.2 Model

3.2.1 Community structure and blockmodels

Consider an undirected network with n nodes and no self-loops or multiple edges, and let A be its adjacency matrix. Such networks can be conceptualized as being generated from a random graph model with

$$A_{ij} \sim Ber(p_{ij}), \tag{3.1}$$

where p is a symmetric probability matrix whose diagonals are zero and off-diagonals are between 0 and 1. For simplicity we assume all probabilities are strictly positive. In the absence of any structure, p consists of $n(n-1)/2$ distinct parameters. Blockmodels characterize the observed community structure in networks as a manifestation of block structure in this p matrix, by representing p_{ij} as functions of a much smaller set of parameters.

Under the K-block SBM, each node belongs to one of K distinct blocks or communities. Let c denote the true community assignment vector with $c_i = a$ if the i^{th} node belongs to the a^{th} community. Then for $i < j$,

$$p_{ij} = P_{c_i c_j}, \tag{3.2}$$

where P is the K-by-K matrix of community link probabilities. Edges are conditionally independent given c and P. Under the SBM, any two nodes in the same community are *stochastically equivalent* as they behave in an identical manner (in a probabilistic sense) towards the rest of the network, in particular they have the same expected degree and degree distribution.

To allow for more realistic degree distributions and different expected degrees, the DCBM adds node-specific degree parameters such that for $i < j$,

$$p_{ij} = \theta_i \omega_{c_i c_j} \theta_j, \tag{3.3}$$

where θ_i and θ_j are the degree parameters for the respective nodes, and ω is the K-

by-K matrix of baseline interaction between communities. Edges are conditionally independent given c, θ, and ω. Identifiability of the parameters is ensured by a constraint of the form $\sum_{i \in \mathcal{N}_a} \theta_i = 1, \ \forall a = 1, \ldots, K$, where $\mathcal{N}_a$ is the set of nodes belonging to community a.

3.2.2 Node popularity

Node popularity is an important aspect of networks and one that is inextricably associated with community structure. The observed popularity of the i^{th} node in the r^{th} community is given by $M_{ir} = \sum_{j \in \mathcal{N}_r} A_{ij}$. The expectation of this quantity is defined as

$$\mu_{ir} = \mathbb{E}[M_{ir}] = \sum_{j \in \mathcal{N}_r} p_{ij}, \tag{3.4}$$

and we will call μ_{ir} as the *popularity* of the i^{th} node in the r^{th} community. In empirical networks, observed popularities of the n nodes in the K communities vary substantially across nodes as well as communities. To realistically model and analyze this behavior, the random graph model must be flexible enough so that node popularities can freely vary across nodes as well as communities. However, both the SBM and the DCBM put unrealistic restrictions on node popularities, and this is the main motivation behind proposing the PABM, which models node popularities in a flexible and realistic manner. We substantiate this by diagnosing some of the restrictions implicit in DCBM.

Putting together definition (3.4) with models (3.2) and (3.3), we see that under the SBM, $\mu_{ir} = n_r P_{c_i r}$ and under the DCBM, $\mu_{ir} = \theta_i \omega_{c_i r}$ where $n_r = |\mathcal{N}_r|$ is the size of community r. Thus the SBM restricts node popularity to be equal for nodes in the same community, while the DCBM restricts node popularity to scale up or down in accordance to degree parameter — which means for two nodes in the same community, the one with higher θ must be uniformly more popular in all communities. First, let $i, j \in \mathcal{N}_r$ and let s_1, s_2 be two communities. Then

$$\frac{\mu_{is_1}}{\mu_{is_2}} = \frac{n_{s_1} P_{rs_1}}{n_{s_2} P_{rs_2}} = \frac{\mu_{js_1}}{\mu_{js_2}}$$

under the SBM, and

$$\frac{\mu_{is_1}}{\mu_{is_2}} = \frac{\theta_i \omega_{rs_1}}{\theta_i \omega_{rs_2}} = \frac{\omega_{rs_1}}{\omega_{rs_2}} = \frac{\mu_{js_1}}{\mu_{js_2}}$$

under the DCBM. Thus under both models, we have the restriction that relative popularity compared across communities must be equal for all nodes in the same community, as the ratio for i must equal that for j. Secondly,

$$\frac{\mu_{is_1}}{\mu_{js_1}} = \frac{n_{s_1} P_{rs_1}}{n_{s_1} P_{rs_1}} = 1 = \frac{\mu_{is_2}}{\mu_{js_2}}$$

under the SBM, and

$$\frac{\mu_{is_1}}{\mu_{js_1}} = \frac{\theta_i \omega_{rs_1}}{\theta_j \omega_{rs_1}} = \frac{\theta_i}{\theta_j} = \frac{\mu_{is_2}}{\mu_{js_2}}$$

under the DCBM. Thus relative popularity compared between nodes in the same community must be equal across all communities, as the ratio does not depend on s_1 and s_2. These restrictions impede node popularities from varying realistically across nodes and communities.

Structural constraints of the SBM and the DCBM also manifest in restrictions in the formation of individual edges, in addition to node popularity. Let $i_1, i_2 \in \mathcal{N}_r$ and $j_1, j_2 \in \mathcal{N}_s$. Under the SBM, the restriction $p_{i_1 j_1} = p_{i_2 j_2}$ is well-known. Suppose under the DCBM, $\theta_{i_1} > \theta_{i_2}$ and $\theta_{j_1} > \theta_{j_2}$, then $p_{i_1 j_1} > p_{i_2 j_2}$ — thus two high-degree nodes are always more likely to be connected than two low-degree nodes. This can be unrealistic, for instance when the communities represent antagonistic factions, and high-degree nodes might represent authority figures in opposing communities, who are very unlikely to connect with each other, whereas two low-degree nodes might be less extremely positioned and therefore be more likely to connect.

3.2.3 The popularity adjusted blockmodel

We propose a new blockmodel, which we call the popularity-adjusted blockmodel (PABM) where for $i < j$,

$$p_{ij} = \lambda_{ic_j} \lambda_{jc_i}, \tag{3.5}$$

where $\lambda_{ir}, 1 \leq i \leq n, 1 \leq r \leq K$, are the popularity scaling parameters and $0 \leq p_{ij} \leq 1$ for all $i < j$. Thus, p_{ij} depends on the popularity of node i in the community to which j belongs, and the popularity of node j in the community to which i belongs. Similar to the identifiability issue with the DCBM as discussed in Karrer and Newman (2011), the PABM also has a scaling identifiability issue. To see this, fix scaling constants $C_{rs} > 0$ for $1 \leq r < s \leq K$ and define $\tilde{\lambda}_{ir} = C_{rs} \lambda_{ir}$ and $\tilde{\lambda}_{js} = \frac{\lambda_{js}}{C_{rs}}$ for all $i \in \mathcal{N}_s, j \in \mathcal{N}_r$. Then the two sets of parameters $\tilde{\lambda}$ and λ result in the same probability matrix p in (3.5). This issue can be resolved by imposing the identifiability constraint that for all pairs r, s of communities,

$$\Lambda_{rs} = \Lambda_{sr} \text{ where } \Lambda_{rs} := \sum_{j \in \mathcal{N}_r} \lambda_{js}. \tag{3.6}$$

Note that under the PABM, $\mu_{ir} = \lambda_{ir}\Lambda_{rc_i}$. We now show that this model does not have the restrictions mentioned in Section 3.2.2. As before, let $i, j \in \mathcal{N}_r$ and let s_1, s_2 be two communities. Then

$$\frac{\mu_{is_1}}{\mu_{is_2}} = \frac{\lambda_{is_1}\Lambda_{s_1r}}{\lambda_{is_2}\Lambda_{s_2r}}, \qquad \frac{\mu_{js_1}}{\mu_{js_2}} = \frac{\lambda_{js_1}\Lambda_{s_1r}}{\lambda_{js_2}\Lambda_{s_2r}},$$

so these ratios depend on i or j, and can vary across nodes in the same community. Also

$$\frac{\mu_{is_1}}{\mu_{js_1}} = \frac{\lambda_{is_1}\Lambda_{s_1r}}{\lambda_{js_1}\Lambda_{s_1r}} = \frac{\lambda_{is_1}}{\lambda_{js_1}}, \qquad \frac{\mu_{is_2}}{\mu_{js_2}} = \frac{\lambda_{is_2}}{\lambda_{js_2}},$$

which depend on s_1 and s_2, and hence can vary across communities. Finally for $i_1, i_2 \in \mathcal{N}_r$ and $j_1, j_2 \in \mathcal{N}_s$, PABM can model the case where i_1, j_1 are high-degree nodes and authority figures in opposing communities while i_2, j_2 are low-degree nodes, but $p_{i_1j_1}$ is smaller than $p_{i_2j_2}$. For this, set λ_{i_1r} and λ_{j_1s} to high values making i_1 and j_1 very popular in their own communities, but λ_{i_1s} and λ_{j_1r} to small values making them unpopular in each other's community. Setting $\lambda_{i_2s} > \lambda_{i_1s}$ and $\lambda_{j_2r} > \lambda_{j_1r}$ ensures $p_{i_1j_1} < p_{i_2j_2}$ without compromising the high popularity of i_1 and j_1 in their own communities. Thus PABM allows more realistic modeling of node popularities and edge probabilities than DCBM.

Heuristically, the degree of a node is a *network-level* feature, and the DCBM can model this feature quite well, by allowing each node to have its own degree parameter. In this DCBM allows a lot more flexibility compared to the classical SBM, since the latter forces expected degree of all nodes in the same community to be equal. However, popularity of a node is a *community-level* feature as the same node can be popular in one community and unpopular in another community, and the DCBM fails to model this feature adequately. DCBM governs the relative behavior of a node in all communities by a single degree parameter, and this forces a high degree node to be relatively popular uniformly across the network, and a low degree node to be uniformly unpopular. To model node popularities in a flexible manner, the random graph model needs parameters for every node-community combination, which is given by the PABM.

It is relevant to note that both the SBM and the DCBM are special cases of PABM. The SBM (3.2) can be expressed in terms of model (3.5) by setting

$\lambda_{ir} = \sqrt{P_{c_i r}}$ where P is a symmetric K-by-K probability matrix, which implies $p_{ij} = \lambda_{ic_j}\lambda_{jc_i} = \sqrt{P_{c_i c_j}}\sqrt{P_{c_j c_i}} = P_{c_i c_j}$ by the symmetry of P. For DCBM, set $\lambda_{ir} = \theta_i\sqrt{\omega_{c_i r}}$ where ω is a symmetric K-by-K community interaction matrix, which implies $p_{ij} = \lambda_{ic_j}\lambda_{jc_i} = \theta_i\sqrt{\omega_{c_i c_j}} \times \theta_j\sqrt{\omega_{c_j c_i}} = \theta_i\omega_{c_i c_j}\theta_j$ by the symmetry of ω.

3.2.4 Detectability of communities

In Section 3.2.1, we mentioned that blockmodels define a block structure for the n-by-n probability matrix p. The model formulations in (3.2), (3.3), and (3.5) of edge probabilities as functions of underlying parameters characterize the community structure for the respective blockmodels. For this community structure to be well-defined, the communities must also be detectable. In this subsection we define the notion of detectability of communities and lay down detectability conditions for the SBM, the DCBM, and the PABM.

We first state our principle of detectability of communities. Suppose we are given K, the number of communities, and p, the edge probability matrix which follows a blockmodel — we also know the formula of p_{ij} for this blockmodel, for instance if the blockmodel is an SBM we know the formula (3.2). Then given any two nodes j_1, j_2, the principle of detectability postulates that by looking at the corresponding columns $\{p_{ij_1}, p_{ij_2}\}_{i=1}^n$ of p, we should be able to determine whether the two nodes belong to the same community or different communities.

For the SBM, if j_1, j_2 belong to the same community, we know from (3.2) that $p_{ij_1} = p_{ij_2}$ for all $i = 1, \ldots, n$. For detectability, we therefore require that whenever j_1, j_2 belong to different communities, there is deviation from the above pattern, allowing us to detect that j_1, j_2 belong to different communities. This can be ensured by the following detectability criterion: for any two distinct communities a and b, there exists some community c such that $P_{ac} \neq P_{bc}$ — in other words, the a^{th} row and the b^{th} row of P must have at least one disagreement. This is a well-known condition used in SBM literature, for example see Bickel and Chen (2009) and Zhao et al. (2012). Under this criterion, we can have the community detection rule for the SBM, that given any two nodes j_1, j_2, if the set of numbers $\{p_{ij_1}/p_{ij_2}\}_{i=1}^n$ are all equal to 1, then j_1 and j_2 belong to the same community, and if $\{p_{ij_1}/p_{ij_2}\}_{i=1}^n$ has one or more values different from 1, then the nodes belong to different communities.

For the DCBM (3.3), if j_1, j_2 belong to the same community, then $p_{ij_1}/p_{ij_2} = \theta_{j_1}/\theta_{j_2}$ which is constant for all $i = 1, \ldots, n$. A natural detectability criterion is that for any two distinct communities a and b, the set of numbers $\{\omega_{ac}/\omega_{bc}\}_{c=1}^{K}$ has at least two distinct values. Under this criterion, we can have the community detection rule for DCBM, that given any two nodes j_1, j_2, if the numbers $p_{ij_1}/p_{ij_2}, i = 1, \ldots, n$, are all equal, then j_1 and j_2 belong to the same community, and if the set $\{p_{ij_1}/p_{ij_2}\}_{i=1}^{n}$ has two or more distinct values, then the nodes belong to different communities.

While the SBM detectability criterion is widely used, we have not seen previous instances of the DCBM detectability criterion in the extensive literature covering DCBM. Therefore to emphasize its relevance, suppose the detectability criterion is not enforced. For a K-block DCBM, suppose there are two distinct communities a and b such that for some $\gamma \neq 1, \omega_{bc} = \gamma\omega_{ac}$ for all $c = 1, \ldots, K$, and suppose there exist $j_1 \in \mathcal{N}_a, j_2 \in \mathcal{N}_b$ such that $\theta_{j_1} = \gamma\theta_{j_2}$. Then $p_{ij_1} = p_{ij_2}$ for all $i = 1, \ldots, n$ which implies j_1 and j_2 are stochastically equivalent, but they belong to different communities, which is counter-intuitive as two nodes that are identical (in a stochastic sense) should belong to the same community. Our detectability criterion for DCBM precludes this possibility.

Finally for the proposed model (3.5), if j_1, j_2 belong to the same community, then $p_{ij_1}/p_{ij_2} = \lambda_{j_1 c_i}/\lambda_{j_2 c_i}$. This ratio changes value only when c_i changes, and hence the set of numbers $\{p_{ij_1}/p_{ij_2}\}_{i=1}^{n}$ can assume at most K distinct values. Therefore, the detectability criterion is that for any j_1, j_2 belonging to different communities, the set of numbers $\{p_{ij_1}/p_{ij_2}\}_{i=1}^{n}$ must take at least $K+1$ distinct values. The community detection rule is that that given any two nodes j_1, j_2, if the set of numbers $\{p_{ij_1}/p_{ij_2}\}_{i=1}^{n}$ assumes K or less distinct values, then j_1 and j_2 belong to the same community, and if $\{p_{ij_1}/p_{ij_2}\}_{i=1}^{n}$ has $K+1$ or more distinct values, then the nodes belong to different communities.

The notion of detectability forms an intuitive link between the statistical task of model fitting through parametric estimation and the machine learning task of community detection. Given a blockmodel structure and under reasonable conditions, we should be able to successfully estimate the parameters of the blockmodel, and therefore satisfactorily estimate the edge probability matrix p. Had we known p exactly, using the detectability rules we could have assigned communities precisely. If the fitted model is a good approximation to the correct model, the detectability rules ensure that community assignment from the estimation process is approximately accurate as well. In the next section, we formulate the likelihood modularity as a tool for this dual task of model fitting and community detection, and in Section 3.4 we formalize this intuition by establishing community detection consistency of the likelihood modularity under PABM.

3.3 Likelihood modularity for PABM

A natural statistical approach for fitting a PABM to a given network is to maximize the likelihood. However, the likelihood function derived from (3.5) does not have closed form solutions for the maximum likelihood estimators. Following the approach used in Karrer and Newman (2011) in the context of an identical issue with the DCBM, we use Poisson likelihood instead of Bernoulli likelihood, which makes MLEs take closed form expressions. This approach of using likelihood modularity based on Poisson likelihood, while keeping the Bernoulli distribution for model definition and theoretical analysis, was adopted earlier in the context of the DCBM in Zhao et al. (2012) — as mentioned in their chapter and the references within it, this has significant practical benefits at the cost of a small approximation error.

We now compute the likelihood, pretending that A is the adjacency matrix of an undirected multigraph with n nodes, possibly including self-edges. Therefore when $i \neq j$, A_{ij} = number of edges between nodes i and j, but the diagonal element A_{ii} = twice the number of self-edges from i to itself. Consider the random graph model where the number of edges between nodes i and j follow a Poisson distribution and $\mathbb{E}[A_{ij}] = p_{ij}$ with p_{ij} having the same expression as (3.5). Note that the expected number of self-edges of node i is given by $\frac{1}{2}\lambda_{ic_i}^2$. Therefore the likelihood is

$$L = \left(\prod_{i<j} \frac{(\lambda_{ic_j}\lambda_{jc_i})^{A_{ij}}}{A_{ij}!} \exp(-\lambda_{ic_j}\lambda_{jc_i}) \right) \times \left(\prod_{i} \frac{(\frac{1}{2}\lambda_{ic_i}^2)^{A_{ii}/2}}{(A_{ii}/2)!} \exp(-\frac{1}{2}\lambda_{ic_i}^2) \right).$$

Ignoring constants, we have the log-likelihood

$$\begin{aligned} l = &\left(\sum_{i<j} A_{ij}\log(\lambda_{ic_j}) + \sum_{i<j} A_{ij}\log(\lambda_{jc_i}) + \sum_{i} A_{ii}\log(\lambda_{ic_i}) \right) \\ &- \left(\sum_{i<j} \lambda_{ic_j}\lambda_{jc_i} + \frac{1}{2}\sum_{i} \lambda_{ic_i}^2 \right) = \sum_{i}\sum_{r} M_{ir}\log(\lambda_{ir}) - \frac{1}{2}\sum_{i}\sum_{j} \lambda_{ic_j}\lambda_{jc_i}, \end{aligned}$$

where observed popularity M_{ir} is defined in Section 3.2.2. To obtain MLEs, fixing i and r,

$$\frac{\partial l}{\partial \lambda_{ir}} = \frac{M_{ir}}{\lambda_{ir}} - \sum_{j \in \mathcal{N}_r} \lambda_{jc_i}$$

and therefore by solving for the left-hand-side quantity as zero, we get

$$M_{ir} = \lambda_{ir}\Lambda_{rc_i} \tag{3.7}$$

as the likelihood equation, where $i = 1, \dots, n$ and $r = 1, \dots, K$. Let $O_{rs} := \sum_{i \in \mathcal{N}_r} M_{is}$ be the number of edges connecting the r^{th} and s^{th} communities. Note that $O_{rs} = O_{sr}$. Summing the likelihood equation (3.7) over $i \in \mathcal{N}_s$,

$$\sum_{i \in \mathcal{N}_s} \lambda_{ir}\Lambda_{rs} = \sum_{i \in \mathcal{N}_s} M_{ir} = O_{rs} \Rightarrow \Lambda_{sr}\Lambda_{rs} = O_{rs}$$

which implies, by imposing constraint (3.6), that $\Lambda_{rs} = \sqrt{O_{rs}}$. Plugging this into the likelihood equation (3.7), the MLE of λ_{ir} (given c) is

$$\hat{\lambda}_{ir} = \frac{M_{ir}}{\sqrt{O_{rc_i}}}. \tag{3.8}$$

Putting these MLEs into the log-likelihood function, we get

$$\hat{l} = \sum_i \sum_r M_{ir} \log(\frac{M_{ir}}{\sqrt{O_{rc_i}}}) - \frac{1}{2} \sum_i \sum_j \frac{M_{ic_j}}{\sqrt{O_{c_j c_i}}} \times \frac{M_{jc_i}}{\sqrt{O_{c_i c_j}}}.$$

Note that the second term can be written as

$$\frac{1}{2} \sum_i \sum_j \frac{M_{ic_j}}{\sqrt{O_{c_j c_i}}} \times \frac{M_{jc_i}}{\sqrt{O_{c_i c_j}}} = \frac{1}{2} \sum_r \sum_s \frac{1}{O_{rs}} \sum_{i \in \mathcal{N}_r} M_{is} \sum_{j \in \mathcal{N}_s} M_{jr} = \frac{1}{2} \sum_r \sum_s O_{rs} = E$$

where E is the total number of edges in the network, which is a constant free from c. Therefore, ignoring constants and multiplying by 2 for notational convenience, the profile likelihood for a given network can be written as

$$Q(c) = 2 \sum_i \sum_r M_{ir} \log\left(\frac{M_{ir}}{\sqrt{O_{rc_i}}}\right) = 2 \sum_i \sum_r M_{ir} \log(M_{ir}) - \sum_r \sum_s O_{rs} \log(O_{rs}). \tag{3.9}$$

As is usual in network literature, we interpret the profile likelihood (3.9) as a *modularity function* and call it likelihood modularity. For a given adjacency matrix, $Q(c)$ is a function of the community assignment c, with the likelihood already maximized (conditional on c) with respect to the λ_{ir}. Therefore $Q(\cdot)$ can be used as a modularity function for community detection, by maximizing this quantity over the set of community assignments. Note that conditional on a community assignment, the corresponding parameter estimates are given by (3.8), from which we can construct the *fitted* PABM using (3.5).

3.4 Consistency of likelihood modularity

We now investigate theoretical properties of the likelihood modularity (3.9) under the PABM. Following Bickel and Chen (2009) and Zhao et al. (2012) we introduce sparsity through the sparsity parameter ρ_n which goes to zero with n, and consider an appropriately scaled version of the modularity function. The parameters λ_{ir} shrink to zero at the rate $\lambda_{ir} = \sqrt{\rho_n}\lambda_{ir}$, which means edge probabilities shrink to zero with n at the rate $p_{ij} = \rho_n p_{ij}$ — this ostensible abuse of notation makes later terminology less cumbersome. We consider the scaled version of the PABM modularity as

$$\begin{aligned} Q(e) &= \frac{2}{n^2\rho_n}\sum_i\sum_r M_{ir}\log\left(\frac{M_{ir}}{\sqrt{O_{e_ir}}}\right) \\ &= \frac{1}{n^2\rho_n}\left(2\sum_i\sum_r M_{ir}\log(M_{ir}) - \sum_r\sum_s O_{rs}\log(O_{rs})\right). \end{aligned}$$

The 'estimated' community assignment is given by

$$\hat{c} = \arg\max_e Q(e), \tag{3.10}$$

where e is any candidate assignment. We want to prove consistency of likelihood modularity under PABM, i.e., show that the proportion of misclustered nodes goes to zero in probability. We define the population version of the likelihood modularity as

$$\tilde{Q}(e) = \frac{2}{n^2\rho_n}\sum_i\sum_r \mu_{ir}(e)\log\left(\frac{\mu_{ir}(e)}{\sqrt{\mathrm{o}_{re_i}(e)}}\right),$$

where $\mu_{ir}(e) = \mathbb{E}[M_{ir}(e)], \mathrm{o}_{rs}(e) = \mathrm{o}_{sr}(e) = \mathbb{E}[O_{rs}(e)]$. We begin by stating assumptions.

Assumption 3.4.1 *The number of communities K is fixed and known. The true assignment c as well as all candidate assignments e have exactly K non-empty communities.*

Assumption 3.4.2 *Sparsity: $\rho_n = \omega(\frac{\log(n)}{\sqrt{n}})$, which implies $\frac{n\rho_n^2}{\log^2(n)} \to \infty$ as $n \to \infty$.*

Assumption 3.4.3 *Identifiability: for any two communities* $1 \leq a, b \leq K, \Lambda_{ab} = \Lambda_{ba}$*, where* Λ_{ab} *is defined in* (3.6).

Assumption 3.4.4 *Detectability: for any two distinct communities* $1 \leq a \neq b \leq K$ *and any two nodes* $j_1 \in \mathcal{N}_a$*,* $j_2 \in \mathcal{N}_b$*, the set* $\left\{ \frac{p_{ij_1}}{p_{ij_2}} \right\}_{i=1}^{n}$ *assumes at least* $K + 1$ *distinct values.*

Assumption 3.4.2 sets the sparsity level required for consistency. Assumptions 3.4.3 and 3.4.4 re-iterate the identifiability condition from Section 3.2.3 and the detectability condition from Section 3.2.4. Our first lemma establishes an uniform concentration bound for the modularity function and its population version under the sparsity condition.

Lemma 3.4.1 *Under Assumptions 3.4.1 and 3.4.2,*

$$\max_{e} |Q(e) - \tilde{Q}(e)| \xrightarrow{\mathbb{P}} 0.$$

A proof of the lemma, as well as proofs of further results, are at the end of this chapter. Lemma 3.4.1 takes care of the random fluctuation arising from the randomness of edge formation. It states that for any community assignment, the observed modularity can be interpreted as a noise-added version of the population modularity with an asymptotically ignorable noise. This validates the use of the population modularity in establishing community detection accuracy in Lemma 3.4.2, in place of the noisy observed modularity.

A somewhat vexing issue that crops up in community detection (and clustering in general) is the identifiability of community (or cluster) labels. We now introduce some notation to deal with this issue. Any candidate assignment $e = (e_1, \ldots, e_n)$ is an n-vector comprising of the K labels where each label appears at least once. Let Π be the symmetric group of all permutations of $\{1, \ldots, K\}$. For $\sigma \in \Pi$, we define $\sigma(e) := (\sigma(e_1), \ldots, \sigma(e_n))$ as the label permutation of e generated by σ, and define

$$\Pi(e) = \{\sigma(e) : \sigma \in \Pi\}$$

as the set of all label permutations of the assignment vector e. Note that for any $e' \in \Pi(e)$, the community assignments e' and e are identical with respect to community detection, and that $Q(e') = Q(e)$, $\tilde{Q}(e') = \tilde{Q}(e)$. We now establish that the true assignment is the *unique* maximizer of the population version of the likelihood modularity.

Lemma 3.4.2 *Under Assumptions 3.4.3 and 3.4.4, $\tilde{Q}(e)$ is 'uniquely' (up to label permutation) maximized at the correct assignment, i.e., for any candidate assignment e,*

$$\tilde{Q}(e) \leq \tilde{Q}(c)$$

where equality holds if and only if $e \in \Pi(c)$.

In light of the heuristic discussion in the last paragraph of Section 3.2, Lemma 3.4.1 establishes the statistical approximation part, that under Assumption 3.4.2, the observed modularity is an asymptotically accurate approximation of its population version. Lemma 3.4.2 provides the second part about clustering accuracy — if we magically had access to the population modularity $\tilde{Q}$, Lemma 3.4.2 ensures that by maximizing $\tilde{Q}$ over all candidate assignments we would have obtained a *perfectly* accurate community assignment. We do not have that access to $\tilde{Q}$, however by Lemma 3.4.1, with a high probability, the observed modularity Q gets very close to $\tilde{Q}$ as the network increases in size. Hence maximizing the observed modularity Q instead of the population modularity $\tilde{Q}$ should give us an *approximately* accurate community assignment for large networks.

We formalize this in Theorem 3.4.1 which is our main result, but first we need a way to quantify this accuracy. For this, we define the error rate of a candidate assignment as the proportion of nodes where the candidate assignment and the true assignment disagree, i.e.,

$$\xi_n(e) = \min_{e' \in \Pi(e)} \frac{1}{n} \sum_{i=1}^{n} I[e'_i \neq c_i], \tag{3.11}$$

where disagreement is minimized over all label permutations of the candidate assignment. Theorem 3.4.1 establishes the consistency of $\hat{c}$ as an estimator of the true assignment c.

Theorem 3.4.1 *Under Assumptions 3.4.1 - 3.4.4,*

$$\xi_n(\hat{c}) \xrightarrow{\mathbb{P}} 0$$

where $\hat{c}$ is defined in (3.10) *and ξ_n is defined in* (3.11).

Remark 3.4.1 A noteworthy aspect of this result is that it is derived under a fully unrestricted model, as we do not put any structural restrictions on the parameters λ_{ir}, other than those necessary for identifiability or detectability. In comparison, for the DCBM, Zhao et al. (2012) established consistency under a restricted model where the degree parameters are latent variables, and their values are restricted to a finite parameter space. Given that the main advantage of DCBM over the classical SBM is that the former allows flexible modeling of expected degrees, this strong structural restriction forcing expected degrees to take values in a finite set appears somewhat counterintuitive. As pointed out by Bickel and Sarkar (2015), this structural assumption of Zhao et al. (2012) effectively characterizes the DCBM as an SBM with a larger number of communities. A similar comment was made earlier by Amini et al. (2013). This structural restriction has also been remarked upon by Jin (2012).

Remark 3.4.2 The main condition leading to the consistency result is Assumption 3.4.2 which requires the expected node degrees to be of order $\omega(\sqrt{n}\log(n))$. This assumption is stronger than those required for consistency of DCBM modularity in Zhao et al. (2012) or that for consistency of SBM modularity in Bickel and Chen (2009). The likelihood modularity is derived from the maximum likelihood estimation of $O(n)$ parameters, which is much larger compared to the finite number of parameters estimated in the chapters cited above. The stronger degree assumption is the cost associated with the benefits of estimating a more complicated model that allows realistic modeling of node popularity. We envisage that as a natural next step, the degree assumption can be relaxed if we impose some constraints on the model or pursue a regularized approach, see the discussion in Section 3.7 for elaboration on this issue.

3.5 Simulation study

We report results from a simulation study that was undertaken to compare the finite sample performance of the PABM modularity defined in (3.9) with the DCBM modularity, which is defined as

$$Q_{DC} = \sum_r \sum_s O_{rs} \log\left(\frac{O_{rs}}{O_r O_s}\right) \tag{3.12}$$

following Karrer and Newman (2011) and Zhao et al. (2012), where $O_r := \sum_{s=1}^{K} O_{rs}$. In Section 3.4 we proved consistency of $\hat{c}$ which was defined in (3.10) as the *global* maximizer of $Q(\cdot)$. However, it is computationally infeasible to perform an exhaustive search over approximately K^n candidate assignments to obtain the global maximizer, and this optimization problem is in principle NP-hard. In practice an appropriate optimization algorithm is used to maximize modularity functions —

e.g., variational methods (Daudin et al., 2008), Kernighan-Lin type algorithms (Karrer and Newman, 2011), pseudo-likelihood algorithms (Amini et al., 2013), to name a few.

In this chapter, we use the so-called extreme points (EP, hereafter) algorithm, which is a state-of-the-art low dimensional optimization algorithm proposed by Le et al. (2015). Briefly, for $K = 2$ the EP algorithm computes the two leading eigenvectors of the adjacency matrix A, and finds the candidate assignments associated with the extreme points of the projection of the cube $[-1, 1]^n$ onto the space spanned by the two leading eigenvectors of A. Let $\mathcal{B}_{can}$ be the set of all such candidate assignments. The modularity function Q (or Q_{DC}) is evaluated on all assignments $b \in \mathcal{B}_{can}$, and the best assignment is defined as the maximizer of Q (or Q_{DC}) over $\mathcal{B}_{can}$, i.e., $\hat{c} := \arg\max_{b \in \mathcal{B}_{can}} Q(b)$ for PABM, and $\hat{c} := \arg\max_{b \in \mathcal{B}_{can}} Q_{DC}(b)$ for DCBM. Some advantages of EP over the competing methods are that EP is free from issues of initialization, that the candidate set $\mathcal{B}_{can}$ consists of only $O(n)$ assignments compared to 2^n for exhaustive search, and that the candidate set $\mathcal{B}_{can}$ is fixed irrespective of the modularity function being optimized, which makes it particularly suitable for comparing performances of various modularity functions. For networks with low degree nodes Le et al. (2015) recommend a regularized version of the algorithm. Interested readers are referred to Le et al. (2015) for more details about the EP algorithm.

We consider networks with two communities, i.e., $K = 2$, and with equal community sizes $n_1 = n_2$. Model parameters are set as $\lambda_{ir} = \alpha\sqrt{\frac{h}{1+h}}$ when $r = c_i$, and $\lambda_{ir} = \beta\sqrt{\frac{1}{1+h}}$ when $r \neq c_i$, where h is the homophily factor. In each community, we designate 50% of the nodes as category 1 and 50% of the nodes as category 2. We set $\alpha = 0.8, \beta = 0.2$ for category 1 nodes and $\alpha = 0.2, \beta = 0.8$ for category 2 nodes. This implies that between a category 1 node and a category 2 node both belonging to community 1 (say), the category 1 node is more popular in community 1 while the category 2 node is more popular in community 2. The homophily factor h determines the magnitude of community structure in the network — the expected number of intra-community edges is h times the expected number of inter-community edges. The edge probability matrix p (see Section 3.2.1) is constructed from the λ_{ir}'s using (3.5). We increase h from 1.5 to 4 in increments of 0.5 to create networks with increasing strength of community structure, and use sample sizes $n = 400, n_1 = n_2 = 200$ and $n = 1000, n_1 = n_2 = 500$. The model design ensures that across the range of h, the expected number of edges in the network stays fixed at around 40,000 for $n = 400$ and 250,000 for $n = 1000$. As h increases, the expected degree of category 1 nodes ranges from 112-140 for $n = 400$

(280-350 for $n = 1000$), while the expected degree of category 2 nodes ranges from 88-60 for $n = 400$ (220-150 for $n = 1000$). Given the high expected degrees, we do not use regularization for EP.

The main motivation behind the PABM is to construct a random graph model that can realistically model node popularity. Therefore, in addition to community detection, it is relevant to compare the accuracy of Q and Q_{DC} in estimating node popularity. For this, we define two measures of estimation error:

$$E_1 = \frac{1}{2\mathcal{E}} \sum_{i=1}^{n} \sum_{r=1}^{K} \left(\hat{\mu}_{ir}(\hat{c}) - \mu_{ir}(c)\right)^2, \tag{3.13}$$

$$E_2 = \frac{1}{2\mathcal{E}} \sum_{i=1}^{n} \sum_{r=1}^{K} \left(\hat{\mu}_{ir}(c) - \mu_{ir}(c)\right)^2, \tag{3.14}$$

where $\mathcal{E}$ is the expected number of edges in the network and $2\mathcal{E} = \sum_{i=1}^{n} \sum_{r=1}^{K} \mu_{ir}(c)$ is the normalizing constant. For any community assignment b, $\hat{\mu}_{ir}(b)$ are the fitted node popularities calculated from the fitted model (PABM or DCBM) given by that assignment. Note that E_1 measures the overall error in estimating node popularities, originating from the combined effects of community detection and parameter estimation. In contrast E_2 measures the community-corrected error in estimating node popularities, originating purely from parameter estimation since we compute estimated node popularities after plugging in the true community assignment.

For each sample size and each value of h, we generated 100 random networks and applied the EP algorithm on each random network to find the optimal assignments given by Q and Q_{DC}. We computed the community detection error ξ_n defined in (3.11) as well as the estimation errors E_1 and E_2 defined in (3.13) and (3.14). The results are in Figure 3.1, averaged across the 100 simulations for each model setting. As expected, both community detection and popularity estimation improved with increasing values of h, as the community structure became increasingly prominent. However across model settings, the error rates for PABM were quite substantially better than those of DCBM, for both community detection and node popularity estimation.

The poor performance of the DCBM is a consequence of its structural constraints outlined in Section 3.1 and Section 3.2.2. Under the DCBM, for two nodes i_1 and i_2 in the same community, if i_1 is more popular than i_2 in community 1, i_1 must also be more popular than i_2 in community 2. Note that in this simulation, for a category 1 node i_1 and a category 2 node i_2 both belonging to community 1 (say), i_1 is more popular than i_2 in community 1, whereas i_2 is more popular than i_1 in community 2. The DCBM fails to model this dynamic behavior leading to poor community detection and inaccurate estimation of node popularities. Even when we remove the effect of community detection and look at E_2, DCBM makes considerable errors in estimating node popularities. It is worth noting that the community detection by DCBM is quite poor (around 15% error) even when $h = 4$, which means there are around four times many intra-community edges than inter-community edges, implying a quite strong community structure.

3.6 Data analysis

We now report the performance of PABM modularity and DCBM modularity in analyzing two networks, the political blogs network and the British MP Twitter network. The political blogs network was compiled by Adamic and Glance (2005) soon after the 2004 U.S. presidential elections, and it consists of blogs about US politics as nodes and hyperlinks between blogs as edges. The blogs were labeled by Adamic and Glance (2005) as either liberal or conservative in the data set, and we consider this as the true community assignment with $K = 2$. This network has been well-studied in networks literature in general and particularly in connection with the DCBM — starting from the original DCBM chapter by Karrer and Newman (2011) to Zhao et al. (2012), Amini et al. (2013), Jin (2012), Bickel and Sarkar (2015), and Le et al. (2015), to name a few. Following the usual practice, we extract the largest connected component and treat it as a simple graph with 1222 nodes and 16714 edges.

The British MP Twitter network was curated by Greene and Cunningham (2013) with nodes as user accounts of 419 British MPs on the social media platform *twitter.com* and three 'layers' of edges between them, namely *mentions*, *follows*, and *retweets*, which are three kinds of interactions that can happen between Twitter users. The true community assignments are given by the political party affiliations of the MPs. There are five political parties into which the 419 nodes are grouped. However, 360 out of 419 MPs belong to only two of these parties, namely Conservative (colloquially called Tories) and Labour Party. We consider only the network spanned by these two parties, i.e., $K = 2$, and the single 'layer' of edges given by retweets. We analyze the largest connected component, which is a network with 329 nodes and 5720 edges.

We analyze both networks using the EP algorithm of Le et al. (2015) as out-

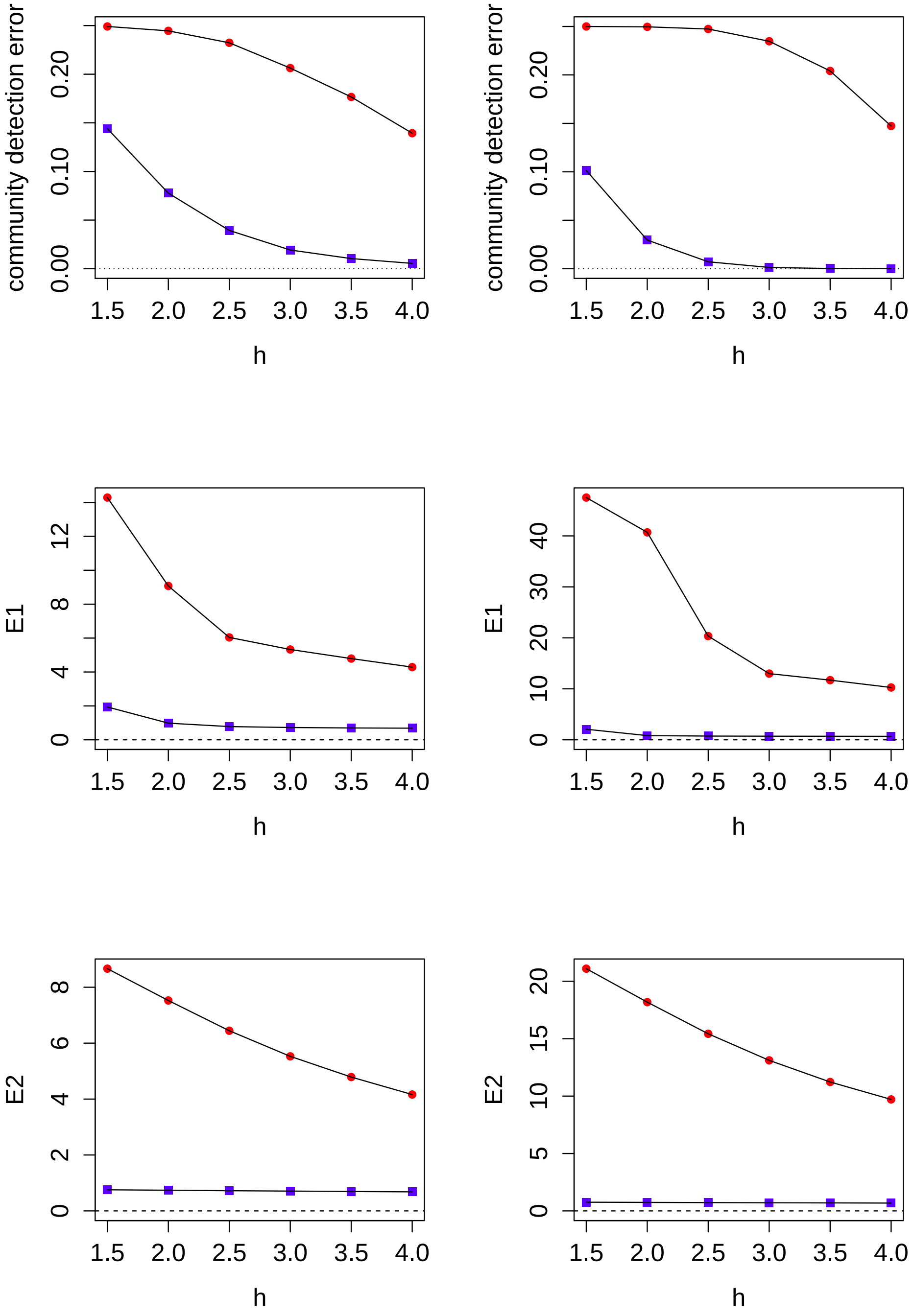


Figure 3.1: Community detection error and popularity estimation error plots from simulation study, where squares represent PABM errors and dots represent DCBM errors. The top row displays community detection errors measured by ξ_n from (3.11), and the middle and bottom rows display estimation errors E_1 and E_2 from (3.13) and (3.14). We use sample sizes $n = 400$ (left) and $n = 1000$ (right). Homophily factor is increased from $h = 1.5$ to $h = 4$ in increments of 0.5. The PABM modularity performs accurate community detection and popularity estimation, whereas results from DCBM are substantially poorer.

lined in Section 3.5. For sparse networks, Le et al. (2015) recommended using a regularization of the form $A + \tau \mathbf{11}'$, where $\tau := \epsilon\lambda_n/n$ and λ_n is the average node degree, n is the number of nodes, and $\epsilon \in (0, 1)$ is a constant. They remarked that the results are insensitive to the value of the tuning parameter ϵ, following the theoretical results of Amini et al. (2013). While this is theoretically true, in practice choosing different values of a tuning parameter can make some differences to the result, and with ad hoc choices reproducibility of results can be difficult. In our data analysis, we therefore considered a range of values $\epsilon = 0.05$ to $\epsilon = 0.95$ in increments of 0.05. Each ϵ corresponds to a set of candidate assignments, say $\mathcal{B}^{\epsilon}_{can}$. We combined these candidate sets across the range of ϵ, and after removing duplicates, considered the superset $\mathcal{B}^{reg}_{can} = \cup_{\epsilon}\mathcal{B}^{\epsilon}_{can}$ as the set of candidate assignments under regularization. As before, we computed Q and Q_{DC} over this candidate set and define the best assignment to be the maximizer of Q (or Q_{DC}) over $\mathcal{B}^{reg}_{can}$, i.e., $\hat{c} := \arg\max_{b \in \mathcal{B}^{reg}_{can}} Q(b)$ for PABM, and $\hat{c} := \arg\max_{b \in \mathcal{B}^{reg}_{can}} Q_{DC}(b)$ for DCBM. We performed both unregularized and regularized versions of EP for both networks and report results from both versions.

As with simulations, for data analysis as well we want to compare the performances with respect to node popularity in addition to community detection. However for observed datasets the true model is unknown, and hence error measures E_1 and E_2 cannot be calculated. Instead, we define the goodness of fit measures:

$$F_1 = \frac{1}{2E}\sum_{i=1}^{n}\sum_{r=1}^{K}\left(\hat{\mu}_{ir}(\hat{c}) - M_{ir}(c)\right)^2, \tag{3.15}$$

$$F_2 = \frac{1}{2E}\sum_{i=1}^{n}\sum_{r=1}^{K}\left(\hat{\mu}_{ir}(c) - M_{ir}(c)\right)^2, \tag{3.16}$$

where E is the observed number of edges, and $2E = \sum_{i=1}^{n}\sum_{r=1}^{K} M_{ir}(c)$ is the normalizing constant. Note that F_1 measures the overall goodness of fit originating from community detection and model fit, while F_2 measures the community-corrected goodness of fit that originates purely from model fit. Results are tabulated in Tables 3.3 and 3.4.

Both modularities perform well with respect to community detection, although the PABM has a slight advantage. However the PABM performs considerably better than the DCBM in terms of fitting node popularities. This poor performance of the DCBM is also a reflection of its structural constraints when it comes to modeling node popularities. Comparing between the unregularized and regularized

		Error from unregularized EP		Error from regularized EP	
Network	Nodes	PABM	DCBM	PABM	DCBM
Political Blogs	1222	4.99% (61)	5.07% (62)	4.99% (61)	5.40% (66)
British MP	329	0.30% (1)	0.61% (2)	0.00% (0)	0.61% (2)

Table 3.3: Community detection error rates (number of misclustered nodes in brackets)

	F_1 from unregularized EP		F_1 from regularized EP		F_2	
Network	PABM	DCBM	PABM	DCBM	PABM	DCBM
Political Blogs	0.06	1.157	0.057	1.155	0.002	1.883
British MP	0.002	0.282	0.002	0.282	0.002	0.284

Table 3.4: Goodness of fit measures for node popularity

versions, there are some small differences for the political blogs network, but results for British MP network are identical for the different version, possibly because the latter is less sparse and hence regularization has little effect.

Finally, we revisit the illustrative nodes from Section 3.1 to see some real implications of the superior popularity fit achieved by the PABM. In Tables 3.5 and 3.6, we revisit Tables 3.1 and 3.2 respectively, but with fitted node popularities under the PABM instead of the DCBM. Clearly the PABM can fit node popularities realistically, and therefore can offer greater insights about the popularity of individual nodes in the networks.

	Observed (Fitted by PABM)			
Name	Community	Liberal Pop	Conservative Pop	Degree
andrewsullivan.com	conservative	58 (59)	85 (84)	143 (143)
blogsforbush.com	conservative	5 (6)	296 (292)	301 (298)
democraticunderground.com	liberal	59 (62)	34 (31)	93 (93)
liberaloasis.com	liberal	169 (169)	2 (1)	171 (170)

Table 3.5: Illustrative nodes for political blogs (regularized EP).

	Observed (Fitted by PABM)			
Name	Community	Conservative Popularity	Labour Pop	Degree
Zac Goldsmith	conservative	46 (46)	25 (25)	71 (71)
Matt Hancock	conservative	68 (67)	3 (3)	71 (70)
Seema Malhotra	labour	0 (0)	88 (88)	88 (88)
Ian Austin	labour	11 (11)	76 (76)	87 (87)

Table 3.6: Illustrative nodes for British MPs. Identities of the nodes of this network were looked up using tweeterid.com. Abbreviations: Comm = community.

3.7 Discussion

This chapter introduces a popularity adjusted blockmodel that can substantially improve modeling of node popularity in networks with community structure, compared to the DCBM. We derive the likelihood modularity for this model and demonstrate its community detection consistency. Using the EP algorithm of Le et al. (2015) we study the performance of this new technique through simulations and analysis of two well-studied networks, and conclude that the new method has substantial advantages over the DCBM.

In Section 3.4 we proved consistency of likelihood modularity under Assumption 3.4.2. In contrast, Zhao et al. (2012) proved consistency of the DCBM likelihood modularity when expected node degrees are of the order $\omega(1)$, which allows for sparser networks — however they assumed strong structural restrictions on the parameters of the DCBM (see Remark 3.4.1). A natural next step for the PABM would be to look for an estimation-cum-community detection method that is consistent under sparser settings than Assumption 3.4.2. We speculate that some form of penalization or regularization of the likelihood function, along with sparsity assumptions on the model parameters, might be a feasible way of achieving this. We consider this as an interesting future direction. It is relevant to note that although in theory consistency of PABM modularity requires stronger assumptions than DCBM modularity, in practice we obtained better fit than the DCBM in Section 3.6 while analyzing the political blogs network, which is a dataset often invoked as a benchmark sparse network (e.g., Zhao et al. (2012), Amini et al. (2013), Jin (2012)).

In this chapter we have assumed K, the number of communities, to be fixed and known. Allowing K to increase with the network size n will require stronger assumptions for consistency, however we hope to be able to accommodate this case under the sparse extension alluded to in the last paragraph. The problem of estimating K from a network under the SBM has recently been investigated in Bickel and Sarkar (2015). Another interesting future direction will be to devise an estimation procedure for K under the PABM.

3.8 Proofs of theoretical results

Proof of Lemma 3.4.1:

We prove this in two steps. First, we establish uniform concentration of $Q(e)$ towards its expected value $\mathbb{E}[Q(e)]$. In the second step, we show that $\mathbb{E}[Q(\cdot)]$ and $\tilde{Q}(\cdot)$ converge (in a deterministic sense) to the same limit.

Step 1. Show that

$$\max_e |Q(e) - \mathbb{E}[Q(e)]| \xrightarrow{\mathbb{P}} 0.$$

The proof of this step relies on Mcdiarmid's inequality (also known as bounded differences inequality, see Theorem 6.2 of Boucheron et al. (2013)), which states: if $f : \mathcal{X}^n \to \mathbb{R}$ satisfies the bounded differences assumption that $\sup_{x_1,\ldots,x_n,x_i'\in\mathcal{X}} |f(x_1,\ldots,x_n) - f(x_1,\ldots,x_{i-1},x_i',x_{i+1},\ldots x_n)| \leq c_i$ for $1 \leq i \leq n$ with constants $c_1,\ldots,c_n > 0$, then

$$\mathbb{P}[| f(\mathbf{x}) - \mathbb{E}[f(\mathbf{x})] |> t] \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right),$$

where $\mathbf{x} = (x_1,\ldots,x_n)$.

For a given assignment e, formulate

$$n^2\rho_n Q(e) = 2\sum_i\sum_r M_{ir}\log(M_{ir}) - \sum_r\sum_s O_{rs}\log(O_{rs})$$

as a function $f(\mathbf{x}) : [0,1]^{n^2} \to \mathbb{R}$ where $\mathbf{x} = \{A_{ij}\}_{1\leq i,j\leq n}$. Note that A_{ij} appears in $f(\mathbf{x})$ only through M_{ie_j} and $O_{e_ie_j}$, and $A_{ij} \in \{0,1\}$, which means the only effect of changing A_{ij} is that M_{ie_j} increases or decreases by 1 and $O_{e_ie_j}$ increases or decreases by 1. So,

$$\begin{aligned}
&|f(A_{11},\dots,A_{nn}) - f(A_{11},\dots,A'_{ij},\dots A_{nn})| \\
&\le 2\max\{|(M_{ie_j}+1)\log(M_{ie_j}+1) - M_{ie_j}\log(M_{ie_j})|, \\
&|(M_{ie_j}-1)\log(M_{ie_j}-1) - M_{ie_j}\log(M_{ie_j})|\} \\
&+\max\{|(O_{e_ie_j}+1)\log(O_{e_ie_j}+1) - O_{e_ie_j}\log(O_{e_ie_j})|, \\
&|(O_{e_ie_j}-1)\log(O_{e_ie_j}-1) - O_{e_ie_j}\log(O_{e_ie_j})|\} \\
&= 2\max\{|\log(M_{ie_j}+1) + M_{ie_j}\log(1+\frac{1}{M_{ie_j}})|, \\
&|\log(M_{ie_j}) + (M_{ie_j}-1)\log(1+\frac{1}{M_{ie_j}-1})|\} \\
&+\max\{|\log(O_{e_ie_j}+1) + O_{e_ie_j}\log(1+\frac{1}{O_{e_ie_j}})|, \\
&|\log(O_{e_ie_j}) + (O_{e_ie_j}-1)\log(1+\frac{1}{O_{e_ie_j}-1})|\} \\
&\le 2\left(\log(n+1)+1\right) + \left(\log(n^2+1)+1\right) \qquad \because x\log(1+1/x)\le 1, \forall x>0 \\
&\le 7\log(n). \qquad \forall n>3
\end{aligned}$$

Thus, for any given e, $Q(e)$ satisfies the bounded differences assumption with $c_{ij} = 7\log(n)$, and $\sum_{i,j} c_{ij}^2 = 49n^2\log^2(n)$. Applying Mcdiarmid's inequality we get that

$$\mathbb{P}[n^2\rho_n \mid Q(e) - \mathbb{E}[Q(e)] \mid > t] \le 2\exp\left(-\frac{2t^2}{49n^2\log^2(n)}\right).$$

Fix $\epsilon > 0$ and let $t = n^2\rho_n\epsilon$. Taking union bound over the K^n possible assignments e,

$$\begin{aligned}
\mathbb{P}[\max_e \mid Q(e) - \mathbb{E}[Q(e)] \mid > \epsilon] &\le 2K^n\exp\left(-\frac{2n^4\rho_n^2\epsilon^2}{49n^2\log^2(n)}\right) \\
&\le 2\exp\left(n\log(K) - \frac{2n^2\rho_n^2\epsilon^2}{49\log^2(n)}\right) \to 0
\end{aligned}$$

where the last convergence follows from Assumption 3.4.2. ♢

Step 2. Show that

$$\max_e |\mathbb{E}[Q(e)] - \tilde{Q}(e)| \to 0.$$

This convergence is a deterministic one, as $\mathbb{E}[Q(e)]$ and $\tilde{Q}(e)$ are non-random numbers. So it suffices to prove that for any given e, $|\mathbb{E}[Q(e)] - \tilde{Q}(e)| \to 0$. Note that

$$\mathbb{E}[Q(e)]-\tilde{Q}(e) = \frac{2}{n^2\rho_n}\sum_i\sum_r\{\mathbb{E}[M_{ir}(e)\log(M_{ir}(e))]-\mathbb{E}[M_{ir}(e)]\log(\mathbb{E}[M_{ir}(e)])\}$$
$$- \frac{1}{n^2\rho_n}\sum_r\sum_s\{\mathbb{E}[O_{rs}(e)\log(O_{rs}(e))]-\mathbb{E}[O_{rs}(e)]\log(\mathbb{E}[O_{rs}(e)])\} \quad (3.17)$$

is comprised of quantities of the form $\mathbb{E}[X\log(X)]-\mathbb{E}[X]\log(\mathbb{E}[X])$. Let X take the form $X = Y_1+\cdots+Y_m$ where $Y_j \sim Ber(p_j)$ and Y_j are independent. The distribution of X is called Poisson binomial distribution or Poisson's binomial distribution (PBD, hereafter) — see Wang (1993) for a review of properties of PBD. Note that $Var[X] = \sum_{j=1}^m p_j(1-p_j) \leq \sum_{j=1}^m p_j = \mathbb{E}[X]$, and hence

$$\begin{aligned}
|\mathbb{E}[X\log(X)]-\mathbb{E}[X]\log(\mathbb{E}[X])| &= |\mathbb{E}[X\log\left(\frac{X}{\mathbb{E}[X]}\right)]| \\
&= \mathbb{E}[X]|\mathbb{E}\left[(Z+1)\log\left(Z+1\right)\right]| \\
&\qquad\qquad (\text{where } Z = \tfrac{X-\mathbb{E}[X]}{\mathbb{E}[X]}) \\
&= \mathbb{E}[X]\mathbb{E}\left[(Z+1)\log\left(Z+1\right)\right] \qquad (3.18)\\
&\leq \mathbb{E}[X]\mathbb{E}[2Z^2] \qquad (3.19)\\
&= 2\frac{Var[X]}{\mathbb{E}[X]} \leq 2. \qquad (3.20)
\end{aligned}$$

Here (3.18) follows from Jensen's inequality, since $x\log(x)$ is a convex function of x and $\mathbb{E}(Z+1)=1$, and the inequality (3.19) follows from the observation that $Z \geq -1$ and

$$\mathbb{E}\left[2Z^2-(Z+1)\log\left(Z+1\right)\right] \geq \mathbb{E}\left[2Z^2-(Z+1)Z\right] = \mathbb{E}\left[Z^2\right]-\mathbb{E}\left[Z\right] = \mathbb{E}\left[Z^2\right] \geq 0.$$

Now, observe that for any e and any i, r, s, M_{ir} and O_{rs} has the form of PBD, hence the bound (3.20) applies to them. Therefore from (3.17), for any e

$$\begin{aligned}
|\mathbb{E}[Q(e)]-\tilde{Q}(e)| &\leq \frac{2}{n^2\rho_n}\sum_{ir}|\{\mathbb{E}[M_{ir}(e)\log(M_{ir}(e))]-\mathbb{E}[M_{ir}(e)]\log(\mathbb{E}[M_{ir}(e)])\}| \\
&+ \frac{1}{n^2\rho_n}\sum_{rs}|\{\mathbb{E}[O_{rs}(e)\log(O_{rs}(e))]-\mathbb{E}[O_{rs}(e)]\log(\mathbb{E}[O_{rs}(e)])\}| \\
&\leq \frac{4nK}{n^2\rho_n}+\frac{2K^2}{n^2\rho_n} \to 0
\end{aligned}$$

by Assumption 3.4.2. ◊

Proof of Lemma 3.4.2:

We first make some preliminary definitions and statements. The well-known log-sum inequality (see Theorem 17.1.2 of Cover and Thomas (2012)) states: for $w_1, \ldots, w_m \geq 0, x_1, \ldots, x_m \geq 0$,

$$\left(\sum_{i=1}^m w_i\right) \log\left(\frac{\sum_{i=1}^m w_i}{\sum_{i=1}^m x_i}\right) \leq \sum_{i=1}^m w_i \log(\frac{w_i}{x_i}) \Rightarrow \sum_{i=1}^m w_i \log(\frac{x_i}{w_i}) \leq (\sum_{i=1}^m w_i) \log\left(\frac{\sum_{i=1}^m x_i}{\sum_{i=1}^m w_i}\right),$$

where equality holds *iff* x_i/w_i are equal for all i. For any candidate assignment e and for communities $1 \leq a, r \leq K$, define the index set $\mathcal{N}_a^{(r)} = \{1 \leq j \leq n : c_j = a, e_j = r\}$ of vertices whose true community is a and candidate community is r. Let $S_{ar} = |\mathcal{N}_a^{(r)}|$ be the size of this set, and $\Lambda_{ab}^{(r)} = \sum_{j=1}^n \lambda_{jb} \mathbb{I}[c_j = a, e_j = r]$ be the 'weights' of this set for $b = 1, \ldots, K$. Here are some properties of $\Lambda_{ab}^{(r)}$ that will be useful for our proof:

1. For any $1 \leq a, b \leq K, \sum_{r=1}^K \Lambda_{ab}^{(r)} = \Lambda_{ab}$.
2. For any $1 \leq i \leq n$ and any $1 \leq r, s \leq K$,
(a) $\mu_{ir}^{(e)} = \mathbb{E}[M_{ir}^{(e)}] = \sum_{j=1}^n \mathbb{E}[A_{ij}]\mathbb{I}[e_j = r] = \sum_{j=1}^n \lambda_{ic_j}\lambda_{jc_i}\mathbb{I}[e_j = r] = \sum_{a=1}^K \lambda_{ia}\Lambda_{ac_i}^{(r)}$.
(b) $\mathrm{o}_{rs}^{(e)} = \mathrm{o}_{sr}^{(e)} = \mathbb{E}[O_{sr}^{(e)}] = \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}[A_{ij}]\mathbb{I}[e_i = s, e_j = r] = \sum_{a,b=1}^K \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}$.

To prove the lemma, we consider $\Delta(e) := \tilde{Q}(e) - \tilde{Q}(c)$ as a function of e, and show that $\Delta(e) \leq 0$ where equality holds if and only if $e \in \Pi(c)$. Ignoring the scaling constant $\frac{2}{n^2\rho_n}$, from properties 2(a) and 2(b) above we can write

$$\tilde{Q}(e) = \sum_{i=1}^n \sum_{r=1}^K \mu_{ir}(e) \log\left(\frac{\mu_{ir}(e)}{\sqrt{\mathrm{o}_{re_i}(e)}}\right) = \sum_{i=1}^n \sum_{a,r=1}^K \lambda_{ia}\Lambda_{ac_i}^{(r)} \log\left(\frac{\sum_{t=1}^K \lambda_{it}\Lambda_{tc_i}^{(r)}}{\sqrt{\sum_{u,v=1}^K \Lambda_{uv}^{(r)}\Lambda_{vu}^{(e_i)}}}\right).$$

When $e = c$, $\mu_{ir}^{(c)} = \lambda_{ir}\Lambda_{rc_i}, \mathrm{o}_{rs}^{(c)} = \Lambda_{rs}\Lambda_{sr} = \Lambda_{rs}^2$ for all $1 \leq i \leq n$ and all $1 \leq r, s \leq K$, so

$$\begin{aligned}\tilde{Q}(c) &= \sum_{i=1}^n \sum_{a=1}^K \lambda_{ia}\Lambda_{ac_i} \log\left(\frac{\lambda_{ia}\Lambda_{ac_i}}{\sqrt{\mathrm{o}_{ac_i}(c)}}\right) = \sum_{i=1}^n \sum_{a=1}^K \lambda_{ia}\Lambda_{ac_i} \log\left(\lambda_{ia}\right) \\ &= \sum_{i=1}^n \sum_{a,r=1}^K \lambda_{ia}\Lambda_{ac_i}^{(r)} \log\left(\lambda_{ia}\right)\end{aligned}$$

by property 1. Therefore,

$$\Delta(e) = \sum_{i=1}^{n} \sum_{a,r=1}^{K} \lambda_{ia} \Lambda_{ac_i}^{(r)} \log\left(\frac{\mu_{ir}(e)}{\lambda_{ia}\sqrt{\mathrm{o}_{re_i}(e)}}\right) = \sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)} \left[\sum_{i\in\mathcal{N}_b^{(s)}} \lambda_{ia} \log\left(\frac{\mu_{ir}(e)}{\lambda_{ia}\sqrt{\mathrm{o}_{rs}(e)}}\right)\right]. \tag{3.21}$$

We first consider the sum under square brackets, and let

$$w_i = \lambda_{ia}, \qquad x_i = \frac{\mu_{ir}(e)}{\sqrt{\mathrm{o}_{rs}(e)}} = \frac{\sum_{t=1}^{K} \lambda_{it}\Lambda_{tb}^{(r)}}{\sqrt{\sum_{u,v=1}^{K} \Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}}.$$

Note that $\sum_{i\in\mathcal{N}_b^{(s)}} w_i = \Lambda_{ba}^{(s)}$, and

$$\sum_{i\in\mathcal{N}_b^{(s)}} x_i = \frac{\sum_{i\in\mathcal{N}_b^{(s)}} \sum_{t=1}^{K} \lambda_{it}\Lambda_{tb}^{(r)}}{\sqrt{\sum_{u,v=1}^{K} \Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}} = \frac{\sum_{t=1}^{K} \Lambda_{tb}^{(r)} \sum_{i\in\mathcal{N}_b^{(s)}} \lambda_{it}}{\sqrt{\sum_{u,v=1}^{K} \Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}} = \frac{\sum_{t=1}^{K} \Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}}{\sqrt{\sum_{u,v=1}^{K} \Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}}.$$

Applying the log-sum inequality, we get

$$\sum_{i\in\mathcal{N}_b^{(s)}}^{n} \lambda_{ia} \log\left(\frac{\sum_{t=1}^{K} \lambda_{it}\Lambda_{tb}^{(r)}}{\lambda_{ia}\sqrt{\sum_{u,v=1}^{K} \Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}}\right) \leq \Lambda_{ba}^{(s)} \log\left(\frac{\sum_{t=1}^{K} \Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}}{\Lambda_{ba}^{(s)}\sqrt{\sum_{u,v=1}^{K} \Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}}\right) \tag{3.22}$$

$$\Rightarrow \text{ From (3.21), } \Delta(e) \leq \sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left(\frac{\sum_{t=1}^{K} \Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}}{\Lambda_{ba}^{(s)}\sqrt{\sum_{u,v=1}^{K} \Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}}\right). \tag{3.23}$$

Next, we write the right-hand side of (3.23) as (term 1 $-$ term 2 $-$ term 3) where

$$\begin{aligned}
\text{term } 1 &= \sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right) \\
&= \frac{1}{2}\sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left((\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)})(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)})\right), \quad \text{(by symmetry)} \\
\text{term } 2 &= \sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left(\Lambda_{ba}^{(s)}\right) = \frac{1}{2}\sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left(\Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}\right), \\
& \hspace{20em} \text{(by symmetry)} \\
\text{term } 3 &= \sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left(\sqrt{\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}\right) \\
&= \frac{1}{2}\sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left(\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}\right).
\end{aligned}$$

Combining the three terms, from (3.23) we have

$$\Delta(e) \leq \frac{1}{2}\sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log\left[\left(\frac{\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right)\left(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)}{\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}\right) \Big/ \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}\right]. \tag{3.24}$$

We now apply the log-sum inequality once more to the right-hand side of (3.24). Defining the index set $\kappa := \{(a,b,r,s) : 1 \leq a,b,r,s \leq K\}$ and letting

$$w_{abrs} = \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}, \qquad x_{abrs} = \frac{\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right)\left(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)}{\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}},$$

$$\begin{aligned}
\sum_{a,b,r,s=1}^{K} \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)} \log&\left[\left(\frac{\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right)\left(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)}{\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}\right) \Big/ \Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}\right] \\
&\leq \left(\sum_{(a,b,r,s)\in\kappa} w_{abrs}\right) \log\left(\frac{\sum_{(a,b,r,s)\in\kappa} x_{abrs}}{\sum_{(a,b,r,s)\in\kappa} w_{abrs}}\right)
\end{aligned} \tag{3.25}$$

by the log-sum inequality. But,

$$\begin{aligned}
\sum_{(a,b,r,s)\in\kappa} x_{abrs} &= \sum_{a,b,r,s=1}^{K} \frac{\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right)\left(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)}{\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}} \\
&= \sum_{r,s=1}^{K} \frac{\sum_{a,b,t,u=1}^{K}\Lambda_{tb}^{(r)}\Lambda_{bt}^{(s)}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}}{\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}} \\
&= \sum_{r,s=1}^{K} \frac{\left(\sum_{a,u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)\left(\sum_{b,t=1}^{K}\Lambda_{tb}^{(r)}\Lambda_{bt}^{(s)}\right)}{\sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}} \\
&= \sum_{r,s,b,t=1}^{K}\Lambda_{tb}^{(r)}\Lambda_{bt}^{(s)} = \sum_{(a,b,r,s)\in\kappa} w_{abrs}.
\end{aligned}$$

Therefore combining (3.24) and (3.25),

$$\Delta(e) \le \frac{1}{2}\left(\sum_{(a,b,r,s)\in\kappa} w_{abrs}\right)\log(1) = 0$$

for any candidate assignment e.

Next, we show that $\Delta(e) = 0$ if and only if $e \in \Pi(c)$. The 'if' part is obvious from the definition of $\Delta(e)$ — see the discussion preceding Lemma 3.4.2. To complete the proof we need to show that $\Delta(e) = 0$ implies $e \in \Pi(c)$.

Suppose $\Delta(e) = 0$. Then the log-sum inequalities in both (3.22) and (3.25) must be equalities, which happens if and only if the ratio w_i/x_i is equal for all i, which implies $w_i/x_i = \sum w_i/\sum x_i$ for all i. Fix any a, b, r, s such that $S_{ar}S_{bs} > 0$. Then from (3.22),

$$\frac{\lambda_{ia}\sqrt{\mathrm{o}_{rs}(e)}}{\mu_{ir}(e)} = \frac{\Lambda_{ba}^{(s)}\sqrt{\mathrm{o}_{rs}(e)}}{\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}}\ , \quad \frac{\lambda_{jb}\sqrt{\mathrm{o}_{rs}(e)}}{\mu_{js}(e)} = \frac{\Lambda_{ab}^{(r)}\sqrt{\mathrm{o}_{rs}(e)}}{\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}}, \quad \forall i \in \mathcal{N}_b^{(s)}, j \in \mathcal{N}_a^{(r)},$$

and multiplying the two equations yields

$$\frac{\lambda_{ia}\lambda_{jb}\mathrm{o}_{rs}(e)}{\mu_{ir}(e)\mu_{js}(e)} = \frac{\Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}\ \mathrm{o}_{rs}(e)}{\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right)\left(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)},$$

while applying the log-sum equality condition on (3.25) yields that

$$\frac{\Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}\ \sum_{u,v=1}^{K}\Lambda_{uv}^{(r)}\Lambda_{vu}^{(s)}}{\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right)\left(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)} = \frac{\Lambda_{ab}^{(r)}\Lambda_{ba}^{(s)}\ \mathrm{o}_{rs}(e)}{\left(\sum_{t=1}^{K}\Lambda_{bt}^{(s)}\Lambda_{tb}^{(r)}\right)\left(\sum_{u=1}^{K}\Lambda_{au}^{(r)}\Lambda_{ua}^{(s)}\right)} = 1.$$

Combining the above two results, for any a, b, r, s such that $S_{ar}S_{bs} > 0$, $\forall i \in \mathcal{N}_b^{(s)}, j \in \mathcal{N}_a^{(r)}$,

$$p_{ij} = \lambda_{ia}\lambda_{jb} = \frac{\mu_{ir}(e)\mu_{js}(e)}{\mathrm{o}_{rs}(e)}. \tag{3.26}$$

We proceed to prove by contradiction that the above condition implies $e \in \Pi(c)$. Suppose there exist distinct communities $a_1 \neq a_2$ such that $S_{a_1 r} > 0, S_{a_2 r} > 0$. Choose any $j_1 \in \mathcal{N}_{a_1}^{(r)}$, any $j_2 \in \mathcal{N}_{a_2}^{(r)}$. By Assumption 3.4.4,

$$\exists\, 1 \leq i_1, \ldots, i_{K+1} \leq n \text{ such that } \frac{p_{i_l j_1}}{p_{i_l j_2}} \neq \frac{p_{i_m j_1}}{p_{i_m j_2}}, \quad \forall\, 1 \leq l \neq m \leq K+1. \tag{3.27}$$

Since there are $K+1$ nodes $\{i_1, \ldots, i_{K+1}\}$ and e has K communities, by the pigeon hole principle there must be $l \neq m$ and some $s \in \{1, \ldots, K\}$ such that $e_{i_l} = e_{i_m} = s$. Let b_l and b_m be the true communities of these nodes, i.e., $c_{i_l} = b_l, c_{i_m} = b_m$. Here b_l and b_m may or may not be equal. Thus, $S_{a_1 r}S_{b_l s} > 0, i_l \in \mathcal{N}_{b_l}^{(s)}, j_1 \in \mathcal{N}_{a_1}^{(r)}$, and $S_{a_2 r}S_{b_m s} > 0, i_m \in \mathcal{N}_{b_m}^{(s)}, j_2 \in \mathcal{N}_{a_2}^{(r)}$. Hence from (3.26)

$$\frac{p_{i_l j_1}}{p_{i_l j_2}} = \frac{\mu_{j_1 s}(e)}{\mu_{j_2 s}(e)}\,,\ \frac{p_{i_m j_1}}{p_{i_m j_2}} = \frac{\mu_{j_1 s}(e)}{\mu_{j_2 s}(e)} \ \Rightarrow\ \frac{p_{i_l j_1}}{p_{i_l j_2}} = \frac{p_{i_m j_1}}{p_{i_m j_2}}$$

which violates (3.27).

Thus, $\Delta(e) = 0$ implies there cannot exist $a_1 \neq a_2$ such that $S_{a_1 r} > 0, S_{a_2 r} > 0$. All communities of e must be non-empty, hence any column of S has exactly one positive entry, all other entries being zero. Therefore S has exactly K non-zero entries. Now, suppose that for some $r_1 \neq r_2$, $S_{ar_1} > 0, S_{ar_2} > 0$. Since the a^{th} row of S has two positive entries, some other row, say b, of S must be empty. But that implies community b is empty in the true community assignment c, which cannot happen. Hence each row of S and each column of S has exactly one non-zero entry. Therefore S is diagonal up to permutation of columns, which implies e is a label permutation of c, i.e., $e \in \Pi(c)$. ♢

Proof of Theorem 3.4.1:

From Lemma 3.4.2, we know that $\xi_n(\hat{c}) > 0 \Rightarrow \tilde{Q}(c) > \tilde{Q}(\hat{c})$ and that $\xi_n(\hat{c}) = 0 \Rightarrow \tilde{Q}(c) = \tilde{Q}(\hat{c})$. So there exists $\delta_n \downarrow 0$ and $\epsilon_n > 0$ such that $\xi_n(\hat{c}) > \delta_n \Rightarrow$

$\tilde{Q}(c) > \tilde{Q}(\hat{c}) + 2\epsilon_n$. Therefore,

$$\begin{aligned}
\mathbb{P}\left[\xi_n(\hat{c}) > \delta_n\right] &\leq \mathbb{P}\left[\tilde{Q}(c) > \tilde{Q}(\hat{c}) + 2\epsilon_n\right] \\
&= \mathbb{P}\left[\{\tilde{Q}(c) > \tilde{Q}(\hat{c}) + 2\epsilon_n\} \cap \{Q(\hat{c}) \geq Q(c)\}\right] \quad \text{(by definition of } \hat{c}\text{)} \\
&\leq \mathbb{P}\left[\{|\tilde{Q}(c) - Q(c)| > \epsilon_n\} \cup \{|\tilde{Q}(\hat{c}) - Q(\hat{c})| > \epsilon_n\}\right] \\
&\leq \mathbb{P}\left[|\tilde{Q}(c) - Q(c)| > \epsilon_n\right] + \mathbb{P}\left[|\tilde{Q}(\hat{c}) - Q(\hat{c})| > \epsilon_n\right] \\
&\to 0 \qquad \text{(by Lemma 3.4.1)}
\end{aligned}$$

which concludes the proof. ♢

CHAPTER 4

THE DEPENDENT RANDOM WEIGHTING

4.1 Introduction

Resampling methodology for dependent data such as time series and spatial data have undergone rapid developments since Künsch (1989) and Liu and Singh (1992) introduced the moving block bootstrap independently. The block-based bootstrap and subsampling methods [Politis and Romano (1994)] have been proved to be very useful nonparametric resampling techniques in the inference of regularly spaced time series and spatial data. The block-based resampling/subsampling methodology, although still applicable and theoretically justified to irregularly spaced time series and spatial data, are practically inconvenient to use. Here we mention Hall (1985), Politis and Romano (1993), Sherman and Carlstein (1994), Sherman (1996), Garcia-Soidan and Hall (1997), Politis et al. (1998), Lahiri (1999), Lahiri et al. (1999), Politis and Sherman (2001), and Nordman and Lahiri (2004), among others for important work along this line. For time series data, the irregularity can occur if there are missing values for a equally-spaced time series, or the time points at which the observation are taken are generated from a one-dimensional point process. In the spatial setting, the irregularly spaced data, which can be in the form of lattice data with an irregular shape of the sampling region, or nonlattice data with spatial locations generated from a spatial point process, are quite common.

For irregularly spaced data, the main difficulty associated with the block-based resampling/subsampling approach is that the partition of sampling region into complete and incomplete blocks requires careful programming efforts and it depends on temporal/spatial configuration to a large extent. This makes the use of block-based methods less automatic so it would be desirable to develop alternative methods whose implementation does not depend on the irregular temporal/spatial configuration. Recently, Shao (2010) proposed the dependent wild bootstrap (DWB, hereafter) for stationary and weakly dependent time series, which has no implementational difficulty when applied to irregular spaced time series. However, the applicability of the DWB is limited to the smooth function model and it cannot be used to approximate the sampling distribution and variance of some other quantities, such as sample median.

In this paper, we propose a new resampling method, called the dependent random weighting (DRW, hereafter), which has wider applicability than the DWB and possesses considerable implementational advantage than the block-based bootstrap and subsampling methods for irregularly spaced dependent data. The random weighting method [Zheng (1987)] has been well studied for iid data and for linear models; see Shao and Tu (1995), Chapter 10 for a detailed introduction. Instead of generating resamples from the data, the random weighting method assigns a random weight to each observation. Random weighting can be regarded as an extension of the Bayesian bootstrap [Rubin (1981)] and a smoothing of Efron's bootstrap. Often the weights can be written as

$$w_i = \frac{Z_i}{\sum_{i=1}^{n} Z_i},\ i = 1, \cdots, n, \tag{4.1}$$

where Z_i are nonnegative iid random variables. So far it seems that the methodological and theoretical developments are confined to the independent data setting. For dependent data, such as time series and spatial data, the original random weighting method, which typically allows the weights to be exchangeable, does not work in general. To capture the dependence in the data, we extends the traditional random weighting to the time series/spatial setting by allowing the Z_i involved in the random weighting method to be dependent, so it is capable of mimicking the dependence in the original series. Section 4.2 describes the DRW and demonstrates the distribution consistency of the DRW estimator for regular and irregular spaced time series. Section 4.3 reports results from simulation studies for irregular time series data (one-dimensional) and spatial data (two-dimensional). Section 4.4 concludes and technical details are gathered in Section 4.5.

4.2 DRW for Time Series

We shall first provide a description of the DRW in the time series context. Suppose we have a stationary p-dimensional time series $(X_t)_{t\in\mathbb{Z}}$ and the parameter of interest is $\theta = T(F)$, where T is a given functional and F is the marginal distribution function of $X_t \in \mathbb{R}^p$. Examples include the mean, marginal variance and quantiles of X_t. The estimator of θ is $\hat{\theta}_n = T(F_n)$, where F_n is the empirical distribution function based on the observations $\{X_{t_j}\}_{j=1}^n$, and $\{t_j\}_{j=1}^n$ are the time points at which the data are observed. In the equally spaced case, $t_j = j$. The randomly weighted empirical distribution function F_n^* is defined as $F_n^*(x) = \sum_{i=1}^n w(t_i)\mathbf{1}(X_{t_i} \leq x)$, where $\{w(t_i)\}_{i=1}^n$ are the random weights. We assume that the weights take the form of (4.1), in particular

$$w(t_i) = \frac{Z(t_i)}{\sum_{i=1}^n Z(t_i)}$$

where $\{Z(t_i)\}$ are a realization from a nonnegative continuous time process $Z(t), t \in \mathbb{R}$.

Assumption 4.2.1 The random variables $\{Z(t_j)\}_{j=1}^n$ are independent of the data, and are a realization of a stationary process with $cov(Z(t_j), Z(t_{j'})) = a\{(t_j - t_{j'})/l\}$, where $a : \mathbb{R} \to [0, 1]$ is continuous, symmetric, and has compact support on $[-1, 1]$. Further assume that $\{Z(t)\}$ is l-dependent.

Several commonly-used windows (kernels) in spectral analysis, such as Bartlett, Parzen and Tukey-Hanning windows, satisfy Assumption 4.2.1 on $a(\cdot)$. The bandwidth parameter l plays a similar role as that in the DWB or the block size in the moving block bootstrap.

Let $\hat{\theta}_n^* = T(F_n^*)$. Then we can approximate the sampling distribution or variance of $\sqrt{n}(\hat{\theta}_n - \theta)$ by the conditional distribution or conditional variance of $\sqrt{n}(\hat{\theta}_n^* - \hat{\theta}_n)S_Z$ given the data, where $S_Z = \mathbb{E}(Z(1))/\sqrt{\text{var}(Z(1))}$ is a scaling factor. It is worth noting that the scaling constant S_Z also comes up in the original random weighting method, and it is in fact possible to select the distribution of $(Z(t))$ so that $S_Z = 1$, as demonstrated in the following example.

Example 4.2.1 *In the equally spaced case, let* $Z_t = (Y_t + c)^2$, *where* $\{Y_t\}_{t=1}^n \sim N(0, \Sigma)$, *where* Σ *is an* $n \times n$ *matrix with* (i, j)*th entry defined as* $\Sigma(i, j) = W((i - j)/l_n)$, *where* W *is a symmetric kernel function. Assuming that* $W(0) = 1$, *then* $\mathbb{E}(Z_1) = \mathbb{E}(Y_1^2) + c^2 = c^2 + 1$ *and* $var(Z_1) = \mathbb{E}(Y_1 + c)^4 - (\mathbb{E}(Z_1))^2 = \mathbb{E}(Y_1^4) + 6c^2\mathbb{E}(Y_1^2) + c^4 - (c^2 + 1)^2 = 4c^2 + 2$. *Setting* $S_Z = 1$, *we get* $4c^2 + 2 = (c^2 + 1)^2$, *which yields* $c^2 = (1 + \sqrt{2})$. *Note that in this case,* $cov(Z_t, Z_{t'}) = 2W^2((t - t')/l_n) + 4c^2W((t - t')/l_n)$. *Same argument applies to the unequally spaced case; see Section 4.3.*

4.2.1 Equally spaced time series

We shall first study the asymptotic properties of the DRW estimator when the time series is evenly spaced, i.e., $t_j = j$. Following Shao (2010), we focus on the framework of the smooth function model, which contains a large class of quantities of interest in time series analysis. Let $\theta = H(\mu)$ where $\mu = \mathbb{E}(X_t)$ and $H : \mathbb{R}^p \to \mathbb{R}$ is a smooth function. Given observations $(X_t)_{t=1}^n$, the estimator is $\hat{\theta}_n = H(\hat{\mu}_n)$, where $\hat{\mu}_n = \bar{\mathbf{X}}_n = n^{-1}\sum_{t=1}^n X_t$. The DRW counterpart of $\hat{\theta}_n$ is $\hat{\theta}^*_{n,DRW} = H(\hat{\mu}^*_{n,DRW})$, where $\hat{\mu}^*_{n,DRW} = \sum_{t=1}^n w_t X_t$. Let $\sigma_n^2 = n\text{var}(\hat{\theta}_n)$ and $\boldsymbol{\nabla}(\mathbf{x}) = \{\partial H(\mathbf{x})/\partial x_1, \partial H(\mathbf{x})/\partial x_2, \cdots, \partial H(\mathbf{x})/\partial x_p\}'$ be the vector of first order partial derivatives of H at $\mathbf{x}$. Denote by $\boldsymbol{\nabla} = \boldsymbol{\nabla}(\boldsymbol{\mu})$ and $\boldsymbol{\Sigma}_\infty = \sum_{k=-\infty}^{\infty} \text{cov}(\mathbf{X}_0, \mathbf{X}_k)$. Under some suitable conditions, we have $\sqrt{n}(\hat{\theta}_n - \theta_0) \to_D N(0, \tau_\infty^2)$, where $\tau_\infty^2 = \boldsymbol{\nabla}'\boldsymbol{\Sigma}_\infty\boldsymbol{\nabla} > 0$.

Denote by $\alpha(k)$ strong mixing coefficients of the process $\mathbf{X}_t$; by $X_{t,i}$ the i-th component of $\mathbf{X}_t$. The following assumptions are needed to state the consistency of the DWB in distribution approximation.

Assumption 4.2.2 Assume that there exists a $\delta \geq 2$ such that $\sum_{j=1}^{\infty} \alpha(j)^{\delta/(2+\delta)} < \infty$ and $\mathbb{E}\|\mathbf{X}_1\|^{2+\delta} < \infty$. Also suppose that $\boldsymbol{\Sigma}_\infty$ is nonsingular.

Assumption 4.2.3 For any $(i_1, i_2, i_3, i_4) \in \{1, 2, \cdots, p\}^4$, we have

$$\sum_{t_1,t_2,t_3=-\infty}^{\infty} |\text{cum}(X_{0,i_1}, X_{t_1,i_2}, X_{t_2,i_3}, X_{t_3,i_4})| < \infty.$$

See Section 3 in Shao (2010) for the discussion of the above assumptions.

Theorem 4.2.1 *Assume that the function H is differentiable in a neighborhood of $\boldsymbol{\mu}$, i.e., $N_H = \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x} - \boldsymbol{\mu}\| \leq \epsilon\}$ for some $\epsilon > 0$, $\sum_{|\boldsymbol{\alpha}|=1} |D^{\boldsymbol{\alpha}} H(\boldsymbol{\mu})| \neq 0$, and the first partial derivatives of H satisfy a Lipschitz condition of order $s > 0$ on N_H. Suppose that Assumptions 4.2.1, 4.2.2, 4.2.3 and $l^{-1} + l/n^{\delta/(2+2\delta)} = o(1)$ hold. Further assume that $Z_t \in \mathcal{L}^{2+\delta}$ for $\delta \geq 2$ (i.e. $E[Z_t^{2+\delta}] < \infty$). Then*

$$\sup_{x\in\mathbb{R}} |P[\sqrt{n}\{H(\bar{\mathbf{X}}_n) - H(\boldsymbol{\mu})\} \leq x] - P^*[\sqrt{n}\{H(\bar{\mathbf{X}}^*_{n,DRW}) - H(\bar{\mathbf{X}}_n)\}S_Z \leq x]| = o_p(1).$$

Remark 4.2.1 The smooth function model framework covers several important parameters and their estimators, for example autocovariances, autocorrelations, autoregressive coefficients etc. The median and other quantiles, however, do not fall in the class of smooth function models. For more detail on the general class of estimators covered under the smooth function model we refer the interested reader to Chapter 4 of Lahiri (2003b), specifically the examples of that chapter.

In general, for approximately linear statistic $T(F_n)$, we can expand $T(F_n)$ around $T(F)$ as $T(F_n) = T(F) + n^{-1}\sum_{t=1}^{n} IF(X_t; F) + R_n$, where $IF(X_t; F)$ is the influence function and R_n is the remainder term. Similarly, we have $T(F_n^*) = T(F)+n^{-1}\sum_{t=1}^{n} w_t IF(X_t; F)+R_n^*$. To show the consistency of $\sqrt{n}(T(F_n^*)-T(F_n))$ as an estimator of $\sqrt{n}(T(F_n) - T(F))$ in terms of distribution approximation, a typical strategy is to find appropriate regularity conditions on T and the weak dependence of X_t to guarantee the asymptotic negligibility of $\sqrt{n}R_n$ and $\sqrt{n}R_n^*$ (conditional on the data), the latter of which may require nontrivial details.

Remark 4.2.2 We encounter some technical difficulty in establishing the consistency for the DRW variance estimator. In particular, it is difficult to obtain a sharp rate for $E\{(1 + U_n)^{-1} - 1\}^2$ (U_n defined in the proof of Theorem 4.2.1; see Section 4.5) which seems necessary to show the variance consistency. The random variable is of the form $\frac{X^2}{(1+X)^2}$ which is hard to control when X is close to -1. In our approach we tried using the power series expansion for $\frac{x}{1+x}$ but that results in a series involving higher moments of the underlying variable, and hence controlling such a quantity would require putting bounds on these higher moments. Such restrictions on higher order moments seem to require stronger assumptions than the standard set of assumptions usually found in the literature. However, from our simulation results in Section 4.3 it appears that variance consistency holds for the model used in our simulations.

Remark 4.2.3 The DRW is closely related to the DWB, which, in a sense, is also a random weighting method [see Section 3 of Shao (2010)]. But the weights of the DWB can be negative and the corresponding bootstrapped measure is not a valid probability measure, which limits its applicability. By contrast, the DRW corresponds to a proper probability measure conditional on the data so it has wider applicability than the DWB. In particular, it can be used to approximate the sampling distribution of sample median and empirical processes for which the DWB is not applicable. The DRW can also be regarded as an extension of the extended tapered block bootstrap [Shao (2009)], where the tapering is applied to the bootstrapped empirical measure corresponding to the moving block bootstrap. However, the extended tapered block bootstrap is still block-based and it encounters implementational difficulty when applied to irregularly spaced data.

4.2.2 Irregularly spaced time series

To allow for irregularly spaced time points, we shall use the theoretical framework in Section 5 of Shao (2010) to study the asymptotic properties of the DRW estimator. In particular, we assume a stochastic sampling design, which was used by Lahiri (2003), Lahiri and Mukherjee (2004) and Lahiri and Zhu (2006) in the study of spatial block bootstrap for irregularly spaced spatial data. Assume that $t_j = \lambda_n v_j$, $j = 1, \ldots, n$, where v_j takes values in $\mathbb{R}_0$ ($\mathbb{R}_0$ is a Borel subset of $(-1/2, 1/2]$ which is the prototype sampling region) and $\{v_j\}_{j=1}^n$ are a realization of the iid random variables $V_1, \ldots, V_n$. This formulation allows a nonuniform design across the region and the expected number of points in two regions of the same size could be different. Assume that there is an underlying 1-dimensional continuous-time stationary process $\{X(t)\}$. Given the observations $\{X(t_j)\}_{j=1}^n$, our interest is in the inference of the mean. Let $\gamma(v_1) = \mathrm{cov}(X(0), X(v_1))$, $C_4(v_1, v_2, v_3) = \mathrm{cum}\{X(0), X(v_1), X(v_2), X(v_3)\}$ denote the autocovariance and the fourth-order cumulant for $v_1, v_2, v_3 \in \mathbb{R}$. Let $\mu = \mathbb{E}(X(t))$ and $\bar{X}_n = n^{-1} \sum_{j=1}^n X(t_j)$. To estimate the distribution and the variance of $\sqrt{n}(\bar{X}_n - \mu)$, we note that the DRW counterpart of $\sqrt{n}(\bar{X}_n - \mu)$ is $\sqrt{n}(\bar{X}^*_{n,DRW} - \bar{X}_n)S_Z$, where $\bar{X}^*_{n,DRW} = \sum_{j=1}^n w(t_j)X(t_j)$. Without loss of generality, we assume that $\{V_n\}_{n\geq 1}$, $\{X(t), t \in \mathbb{R}\}$ and the bootstrap variables $\{Z(t_j), t \in \mathbb{R}\}$ are all defined on a common probability space $(\Omega, \mathcal{F}, P)$. Let P_V denote the joint probability distribution of the sequence of iid random variables $V_1, V_2, \ldots, V_n$ with density $\eta(v)$, $v \in R_0$. We shall use $\mathbb{E}_V$ (var_V) to denote the expectation (variance) with respect to the joint distribution P_V; use $\mathbb{E}_{X|V}$ ($\mathrm{var}_{X|V}$) to denote the conditional expectation (variance) with respect to P_X (i.e., the joint probability distribution for $\{X(t), t \in \mathbb{R}\}$) given $\{V_n\}_{n\geq 1}$. Following Shao (2010), we assume the following assumption on the sampling region R_0 and sampling density $\eta(\cdot)$.

Assumption 4.2.4 Define R_0 to be a Borel subset of $(-1/2, 1/2]$ containing an open neighborhood of the origin such that for any sequence of positive real numbers $a_n \to 0$, the number of cubes of the scaled lattice $a_n\mathbb{Z}$ which intersect R_0 and R_0^c is $O(1)$ as $n \to \infty$.

Assumption 4.2.5 The pdf $\eta(x)$ is continuous, everywhere positive with support $\bar{R}_0$ and $\int_{s\in R_0} \eta(s)ds = 1$.

Denote by $\iota = \int_{s\in R_0} \eta^2(s)ds$. Lahiri (2003) showed that depending on the magnitude of $\kappa := \lim_{n\to\infty} n/\lambda_n$, this formulation accommodates both pure-increasing-domain asymptotics (i.e., $\kappa < \infty$) and mixed-increasing-domain asymptotics (i.e., $\kappa = +\infty$). Let $\xi_n = \mathrm{var}(\bar{X}_n)$. Lemma 5.2 of Lahiri (2003) implies that under appropriate conditions, we have that (i) if $\kappa \in (0, \infty)$, then $n\xi_n \to \gamma(0) + \kappa\iota \int_{\mathbb{R}} \gamma(s)ds$, a.s. (P_V); (ii) if $\kappa = \infty$, then $\lambda_n \xi_n \to \iota \int_{\mathbb{R}} \gamma(s)ds$, a.s. (P_V). Here a.s. (P_V) means that the result holds with probability one under P_V, i.e., for almost all realizations of the sequence $\{V_n\}_{n\geq 1}$. In Lahiri (2003), the distribution of $\bar{X}_n$ is regarded as conditional distribution given $\{V_n\}_{n\geq 1}$ and ξ_n is regarded as a function of the randomly sampled locations. Whereas in our treatment, we view ξ_n as an unknown quantity, where the randomness due to $\{V_n\}_{n\geq 1}$ has been removed by the expectation. The following theorem states the distribution consistency of the DRW estimator.

Theorem 4.2.2 *Suppose that Assumptions 4.2.1, 4.2.4 and 4.2.5 hold. Assume that $l_n/\sqrt{n} + l_n/\lambda_n = o(1)$,*

$$\int_{\mathbb{R}} |\gamma(v)|dv \quad < \quad \infty, \tag{4.2}$$

$$\text{and} \int_{\mathbb{R}^3} |C_4(v_1, v_2, v_3)|dv_1 dv_2 dv_3 \quad < \quad \infty. \tag{4.3}$$

Further assume that $Z(t) \in \mathcal{L}^4$. We have that (i) if $\kappa \in (0, \infty)$, then

$$\sup_{x\in\mathbb{R}} |P[\sqrt{n}(\bar{X}_n - \mu) \leq x] - P^*[\sqrt{n}(\bar{X}^*_{n,DRW} - \bar{X}_n)S_Z \leq x]| = o_p(1),$$

and that (ii) if $\kappa = \infty$, then

$$\sup_{x\in\mathbb{R}} |P[\sqrt{\lambda_n}(\bar{X}_n - \mu) \leq x] - P^*[\sqrt{\lambda_n}(\bar{X}^*_{n,DRW} - \bar{X}_n)S_Z \leq x]| = o_p(1).$$

Remark 4.2.4 Lahiri and Zhu (2006) showed that a naive application of block bootstrap, called DSSBB, is not suitable for irregularly spaced data when the spatial sampling density is non-uniform. The DRW does not suffer from the same problem. To quote from their paper, "*The failure of the DSSBB method seems to be an artifact of the interaction between the nonuniform design density and of the additional randomness in the data-site-shifted blocks induced by the random locations of the sampling sites.*" In the case of DRW, resampling takes place by assigning random weights to the data points without shifting their locations. These random weights are independent of the data and spatially correlated to reflect the dependence in the data. Therefore our resampling scheme is free from the interaction alluded to by Lahiri and Zhu (2006) in their explanation of why the DSSBB fails.

4.3 Simulation results

In this section, we investigate the finite sample performance of DRW and its competitors for irregular time series and spatial data under the framework of stochastic sampling design. Let $R_0 = (-1/2, 1/2]$, sample size $n = 200$ and $\lambda_n = 18$ or 36. The time points $\{t_j\}_{j=1}^n$ are generated by taking iid draws from truncated N(0,1) density function over R_0 and multiplying by the scaling constant λ_n. Given the sampled time points, we then generate the observations $\{X(t_j)\}_{j=1}^n$ from a one-dimensional zero-mean Gaussian process with exponential covariance function $\gamma(z) = \exp(-\rho|z|),\ z \in \mathbb{R}$, where $\rho = 0.5$, 1 and 2. The random weights are generated by following Example 4.2.1 and letting $S_Z = 1$ and W to be the Bartlett kernel.

Table 4.1 below shows the normalized mean squared error and the empirical coverages in percentage for the bootstrap approximation of the variance and distribution of the sample median based on the grid based block bootstrap [Lahiri and Zhu (2006)] and DRW. For the variance estimator, let the true variance be σ_n and let $\sigma_n^{(j)}$ be its estimate based on 1000 bootstrap samples for the j^{th} replicate, where $j = 1, 2, \ldots, 1000$ because 1000 monte carlo replications are used. Then the normalized MSE is calculated as $\frac{1}{1000}\sum_{j=1}^{1000}(\frac{n\sigma_n^{(j)}}{n\sigma_n} - 1)^2$. We calculate MSE for the DRW variance estimator, even though we have not demonstrated its asymptotic consistency. It can be seen that in terms of smallest normalized MSE or best empirical coverage (boxed), DRW typically performs at par with GBBB and sometimes marginally outperforms GBBB. For larger block sizes, DRW typically outperforms GBBB by a substantial margin. Larger ρ corresponds to smaller MSE and superior coverage, which is expected because larger ρ implies weaker dependence. Moreover, the MSE decreases and coverage probability improves as λ_n decreases. For the mean case (Table 4.2), DRW, DWB, and GBBB perform similarly in terms of best result in each row (boxed values). For larger block sizes, DRW typically outperforms GBBB by a substantial margin as before, and marginally outperforms DWB. Note that the implementation of grid-based block bootstrap is rather complicated, whereas the DRW can be easily programmed and is also computationally less expensive.

Conceptually the extension of the DRW to irregularly spaced spatial data is straightforward, but technically it seems nontrivial and quite challenging. Here we shall provide a description and some finite sample results for DRW applied to spatial data. Given n spatial locations $\{s_i\}_{i=1}^n$, the observations are assumed to be $\{X(s_i)\}_{i=1}^n$. We shall assume that the observations are from a stationary random field in $\mathbb{R}^2$ (for the sake of simplicity) and the locations can be in a lattice with fixed spacing or irregularly spaced. Let $\theta = T(F)$ be the parameter of interest, where F is the marginal cdf of $X(s)$ and T is a given functional. This framework includes spatial mean and quantiles. Let F_n be the empirical cdf based on $\{X(s_i)\}_{i=1}^n$. Then the sampling distribution or variance of $\sqrt{n}(T(F_n) - T(F))$ can be approximated by the random weighting counterpart $\sqrt{n}(T(F_n^*) - T(F_n))$, with the random weighted empirical cdf defined by

$$F_n^*(x) = \sum_{j=1}^n w(s_j)\mathbf{1}(X_j \leq x), \text{ where } w(s_j) = \frac{Z(s_j)}{\sum_{j=1}^n Z(s_j)}.$$

Here $\{Z(s_j)\}_{j=1}^n$ are nonnegative random variables that are independent of the data, and spatially correlated. In particular, we can mimic Example 4.2.1 and let $Z(s_j) = (Y(s_j) + c)^2$, where $\{Y(s_j)\}_{j=1}^n \sim N(0, \Sigma)$, and Σ is a $n \times n$ matrix with (i,j)th entry defined as $\Sigma(i,j) = W((\|s_i - s_j\|)/l_n)$, where W is a kernel function and $\|s\| = \sqrt{s_{(1)}^2 + s_{(2)}^2}$ for any $s = (s_{(1)}, s_{(2)}) \in \mathbb{R}^2$. Again c can be chosen such that the scaling constant $S_Z = 1$. For spatial data, both subsampling and block based bootstrap implicitly have some requirements on the sampling design (e.g., they may not work well when the sampling design is very heterogeneous) and their implementation is quite involved in the irregularly spaced case. By contrast, the DWB does not involve block sampling but rather generate random and spatially correlated weights to the data. The irregular configuration does not really cause any difficulty in its implementation.

Following the discussion above, we also performed simulations for the two-dimensional case, where ρ is fixed at 1, and two sample sizes, $n = 200, 400$ are used. The normalized MSE for the variance estimator and coverage rates for bootstrap-based intervals for sample mean and sample median are shown in Tables 4.3 and 4.4 respectively. Similar to the one-dimensional case, we observe that in terms of best result in each row (boxed values), DRW and GBBB (and DWB for the mean case) perform similarly, while for larger block sizes, the DRW typically outperforms GBBB by a substantial margin and DWB by a small margin.

4.4 Conclusion

In this paper, we proposed a new resampling method, the dependent random weighting, for time series and briefly mention its extension to spatial data. The main attraction of this new method lies in its adaptiveness to the irregularity of temporal or spatial configurations as its implementation in the irregularly spaced case is the same as regularly spaced case, unlike the block-based bootstrap or subsampling methods. Under suitable conditions, we proved its consistency in distribution approximation for both equally and unequally spaced time series. It is expected that additional theoretical results, such as consistency of bootstrapping empirical processes in both equally and unequally spaced time series (see Bühlmann (1994), Naik-Nimbalkar and Rajarshi (1994) and Peligrad (1998) among others for consistency of block-based bootstrap), and consistency in distribution approximation in the spatial case, can hold under certain regularity conditions. However, this may require a very technical analysis and we leave this for future work. Another topic that is worthy of investigation is the optimal choice of bandwidth parameter l_n for a given kernel function. Also it is of interest to see if one can borrow the recently popular fixed-b asymptotics [Kiefer and Vogelsang (2005)] and calibrate the bootstrap based inference; see Shao and Politis (2013) for a recent attempt along this direction.

4.5 Proofs of theoretical results

Proof of Theorem 4.2.1: Let $\Phi(\mathbf{x};\mathbf{\Sigma}_\infty)$ be the distribution function of $N(\mathbf{0},\mathbf{\Sigma}_\infty)$ on $\mathbb{R}^p$. We first show that

$$\sup_{\mathbf{x}\in\mathbb{R}^p} |P\{\sqrt{n}(\bar{\mathbf{X}}_n-\boldsymbol{\mu})\le\mathbf{x}\} - P^*\{\sqrt{n}(\bar{\mathbf{X}}^*_{n,DRW}-\bar{\mathbf{X}}_n)S_Z\le\mathbf{x}\}| = o_p(1). \quad (4.4)$$

Since $\sqrt{n}(\bar{\mathbf{X}}_n-\boldsymbol{\mu})\to_D N(\mathbf{0},\mathbf{\Sigma}_\infty)$ under Assumption 4.2.2, it follows from a multivariate version of Polyā's theorem (Bhattacharya and Rao 1986) that

$$\sup_{\mathbf{x}\in\mathbb{R}^p} |P\{\sqrt{n}(\bar{\mathbf{X}}_n-\boldsymbol{\mu})\le\mathbf{x}\} - \Phi(\mathbf{x};\mathbf{\Sigma}_\infty)| = o(1).$$

Then (4.4) follows if we can show that

$$\sup_{\mathbf{x}\in\mathbb{R}^p} \left|P^*\{\sqrt{n}(\bar{\mathbf{X}}^*_{n,DRW}-\bar{\mathbf{X}}_n)S_Z\le\mathbf{x}\} - \Phi(\mathbf{x};\mathbf{\Sigma}_\infty)\right| = o_p(1). \quad (4.5)$$

To this end, we shall first establish the relation between the DRW estimator and the DWB estimator introduced in Shao (2010). For DWB,

$$T^*_{n,DWB} = \sqrt{n}(\bar{X}^*_{n,DWB} - \bar{X}_n) = \frac{1}{\sqrt{n}}\sum_{t=1}^{n}(X_t - \bar{X}_n)\delta_t,$$

where δ_t is independent of $\mathcal{X}_n, E(\delta_t) = 0, Var(\delta_t) = 1$, and $Cov(\delta_t, \delta_{t'}) = a(\frac{t-t'}{l})$. For DRW,

$$\begin{aligned} T^*_{n,DRW} &= \sqrt{n}(\bar{X}^*_{n,DRW} - \bar{X}_n)S_Z = \sqrt{n}\sum_{t=1}^{n}\frac{Z_t}{\sum_{t=1}^{n} Z_t}(X_t - \bar{X}_n)S_Z \\ &= \frac{1}{\sqrt{n}}(\frac{1}{n}\sum_{t=1}^{n} Z_t)^{-1}\sum_{t=1}^{n}[(Z_t - E(Z_1))(X_t - \bar{X}_n)]\frac{E(Z_1)}{\sqrt{Var(Z_1)}} \\ &= \left(\frac{1}{\sqrt{n}}\sum_{t=1}^{n}(X_t - \bar{X}_n)\frac{Z_t - E(Z_1)}{\sqrt{Var(Z_1)}}\right)\left(\frac{\frac{1}{n}\sum_{t=1}^{n} Z_t}{E(Z_1)}\right)^{-1} \\ &= \left(\frac{1}{\sqrt{n}}\sum_{t=1}^{n}(X_t - \bar{X}_n)\delta_t\right)\left(\frac{\frac{1}{n}\sum_{t=1}^{n} Z_t}{E(Z_1)}\right)^{-1} = T^*_{n,DWB}\left(1 + U_n\right)^{-1}, \end{aligned}$$

where $\delta_t = \frac{Z_t - E(Z_1)}{\sqrt{Var(Z_1)}}$ and $U_n = \frac{1}{nS_Z}\sum_{t=1}^{n}\delta_t$. Note that

$$E^*\left[(\sum_{t=1}^{n}\delta_t)^2\right] = \sum_{t_1,t_2=1}^{n} E\left[\delta_{t_1}\delta_{t_2}\right] \le 2\sum_{1\le t_1\le t_2\le n}\left|E\left[\delta_{t_1}\delta_{t_2}\right]\right| = O(nl)$$

in view of the fact that under l-dependence, $E\left[\delta_{t_1}\delta_{t_2}\right] \neq 0$ only when $|t_1 - t_2| \le l$. Thus

$$E^*[U_n^2] = \frac{1}{n^2S_Z^2}E^*\left[(\sum_{t=1}^{n}\delta_t)^2\right] = O\left(\frac{l}{n}\right) \to 0 \text{ as n} \to \infty,$$

which implies $U_n \to^P 0$. Further note that for $|x| < \frac{1}{2}, |\frac{x}{1+x}| < 2|x|$ and hence for any $0 < \epsilon < \frac{1}{2}$,
$1 \leftarrow P\left[|U_n| \le \frac{\epsilon}{2}\right] \le P\left[|\frac{1}{1+U_n} - 1| \le \epsilon\right]$, i.e.,

$$(1 + U_n)^{-1} \to^P 1.$$

Hence (4.5) holds by conditional Slutsky's theorem (see Lemma 4.1 of Lahiri (2003b)) and the fact that

$$T^*_{n,DWB} = \frac{1}{\sqrt{n}}\sum_{t=1}^{n}(X_t - \bar{X}_n)\delta_t \to_D N(0, \mathbf{\Sigma}_\infty) \tag{4.6}$$

in probability conditional on the data, the latter of which has been shown in Shao (2010). Finally, our conclusion follows from the argument in the proof of theorem 4.1 of Lahiri (2003b). We omit the details here. $\diamondsuit$

Proof of Theorem 4.2.2: We prove case (i) only, as case (ii) can be dealt with in a similar fashion. For case (i), following the argument in the proof of Theorem 4.2.1, we can write

$$T^*_{n,DRW} = \sqrt{n}(\bar{X}^*_{n,DRW} - \bar{X}_n)S_Z \quad = \left(\tfrac{1}{\sqrt{n}}\textstyle\sum_{j=1}^n (X(t_j) - \bar{X}_n)\delta(t_j)\right)(1+U_n)^{-1} \\ = \tfrac{T^*_{n,DWB}}{1+U_n}$$

where $\delta_{t_j} = \frac{Z_{t_j} - E(Z_1)}{\sqrt{Var(Z_1)}}$ and $U_n = \frac{1}{nS_Z}\sum_{j=1}^n \delta(t_j)$. We want to show that $E^*[U_n^2] = E[U_n^2] \to 0$. Note that

$$E\left[(\sum_{j=1}^n \delta(t_j))^2\right] = E_V\left[E[(\sum_{j=1}^n \delta(t_j))^2|V]\right] = E_V\left[\sum_{j,j'=1}^n a\left(\frac{t_j - t_{j'}}{l_n}\right)\right].$$

For $j = j'$, clearly $E_V\left[a\left(\frac{t_j - t_{j'}}{l_n}\right)\right] = 1$. For $j \neq j'$,

$$E_V\left[a\left(\frac{t_j - t_{j'}}{l_n}\right)\right] = \int_{R_0^2} a\left(\frac{\lambda_n(x-y)}{l_n}\right)\eta(x)\eta(y)dxdy.$$

Let $R_1 = \{x - y : x \in R_0, y \in R_0\}$ and for $z \in R_1$, let $R(z) = (R_0 + z) \cap R_0$. Then, the above integral equals

$$\int_{R_1}\int_{x\in R(z)} a\left(\frac{\lambda_n z}{l_n}\right)\eta(x)\eta(x-z)dxdz \quad = \tfrac{l_n}{\lambda_n}\int_{\frac{\lambda_n}{l_n}R_1}\int_{x\in R(\frac{tl_n}{\lambda_n})} a(t)\eta(x)\eta(x - \tfrac{tl_n}{\lambda_n})dxdt \\ = \tfrac{l_n}{\lambda_n}I_n.$$

Since $a(\cdot)$ has compact support on $[-1,1]$, and for $|t| \leq 1$, $\frac{tl_n}{\lambda_n} = o(1)$, so it follows from the continuity of $\eta(\cdot)$ that

$$\limsup |I_n| \leq \limsup \int_{|t|\leq 1}\int_{x\in R(\frac{tl_n}{\lambda_n})} \eta(x)\eta(x - \frac{tl_n}{\lambda_n})dxdt = \iota < \infty,$$

and hence,

$$E(U_n^2) = E_V\left[\frac{1}{n^2}\sum_{j,j'=1}^n a\left(\frac{t_j - t_{j'}}{l_n}\right)\right] = \frac{1}{n^2} \times O\left(n + n^2\frac{l_n}{\lambda_n}\right) = o(1)$$

which implies that $\frac{1}{U_n+1} \rightarrow^P 1$ similar to the regular time series case. The conclusion follows from the consistency of the DWB (see Theorem 5.2 of Shao (2010)) and an application of conditional Slutsky's theorem (Lemma 4.1. of Lahiri (2003b)). The proof for case (ii) follows in a similar fashion, and we skip the details. $\diamondsuit$

Table 4.1: Top panel: the normalized MSEs for the bootstrap variance estimators of $nvar[median(x_1, \cdots, x_n)]$ using (a) The grid based block bootstrap (b) The dependent random weighting. The box for each row indicates the smallest normalized MSE among $l_n = 1, \cdots, 10$. Bottom panel: the empirical coverage (in percent) for the bootstrap-based confidence intervals of the median using (a) and (b). The box for each row indicates the best coverage among $l_n = 1, \cdots, 10$ (Nominal level is 95%).

							l					
λ_n	ρ		1	2	3	4	5	6	7	8	9	10
18	0.5	(a)	0.57	0.48	$\boxed{0.47}$	0.50	0.52	0.54	0.57	0.62	0.73	0.72
		(b)	0.59	0.48	$\boxed{0.45}$	0.46	0.47	0.49	0.50	0.51	0.52	0.52
	1	(a)	0.38	$\boxed{0.34}$	0.38	0.43	0.49	0.50	0.52	0.57	0.83	0.75
		(b)	0.39	$\boxed{0.32}$	0.34	0.37	0.39	0.41	0.43	0.44	0.45	0.45
	2	(a)	$\boxed{0.27}$	0.29	0.37	0.40	0.45	0.44	0.43	0.49	0.67	0.62
		(b)	$\boxed{0.27}$	0.28	0.31	0.35	0.37	0.39	0.41	0.41	0.42	0.43
36	0.5	(a)	0.51	0.38	$\boxed{0.35}$	0.35	0.37	0.41	0.43	0.46	0.51	0.53
		(b)	0.53	0.39	0.34	$\boxed{0.33}$	0.34	0.35	0.36	0.37	0.38	0.40
	1	(a)	0.32	$\boxed{0.25}$	0.27	0.29	0.33	0.37	0.39	0.40	0.45	0.46
		(b)	0.33	$\boxed{0.25}$	0.25	0.27	0.29	0.30	0.32	0.34	0.34	0.35
	2	(a)	$\boxed{0.18}$	0.19	0.22	0.25	0.28	0.33	0.33	0.35	0.40	0.41
		(b)	0.18	$\boxed{0.17}$	0.20	0.22	0.24	0.27	0.27	0.29	0.30	0.31
18	0.5	(a)	64	71	$\boxed{72}$	71	69	67	63	59	65	59
		(b)	63	70	$\boxed{72}$	72	71	71	70	68	68	67
	1	(a)	76	80	$\boxed{81}$	79	76	75	72	66	70	63
		(b)	75	79	$\boxed{81}$	81	79	79	77	75	74	74
	2	(a)	85	$\boxed{86}$	84	82	80	78	76	70	73	66
		(b)	85	$\boxed{86}$	86	85	83	82	81	80	79	78
36	0.5	(a)	68	77	79	80	$\boxed{81}$	79	79	76	76	74
		(b)	67	76	79	80	$\boxed{82}$	82	82	81	80	80
	1	(a)	81	85	$\boxed{88}$	88	87	86	85	84	80	80
		(b)	80	85	86	$\boxed{87}$	87	87	86	86	85	84
	2	(a)	88	$\boxed{90}$	89	89	88	85	85	83	80	80
		(b)	88	$\boxed{90}$	90	89	89	88	87	87	86	86

Table 4.2: Top panel: the normalized MSEs for the bootstrap variance estimators of $nvar(\bar{x}_n)$ using (a) The dependent wild bootstrap (b) The dependent random weighting (c) The grid based block bootstrap. The box for each row indicates the smallest normalized MSE among $l = 1, \cdots, 10$. Bottom panel: the empirical coverage (in percent) for the bootstrap-based confidence intervals of the mean using (a), (b) and (c). The box for each row indicates the best coverage among $l_n = 1, \cdots, 10$ (Nominal level is 95%).

			l									
λ_n	ρ		1	2	3	4	5	6	7	8	9	10
18	0.5	(a)	0.67	0.54	0.48	$\boxed{0.47}$	0.47	0.48	0.50	0.52	0.54	0.55
		(b)	0.69	0.56	0.51	0.49	$\boxed{0.48}$	0.49	0.49	0.50	0.51	0.52
		(c)	0.67	0.55	$\boxed{0.48}$	0.50	0.50	0.53	0.57	0.62	0.67	0.73
	1	(a)	0.45	0.33	$\boxed{0.32}$	0.34	0.37	0.41	0.44	0.46	0.48	0.50
		(b)	0.48	0.36	$\boxed{0.33}$	0.34	0.35	0.38	0.40	0.41	0.43	0.44
		(c)	0.45	0.34	$\boxed{0.33}$	0.36	0.40	0.42	0.47	0.55	0.62	0.67
	2	(a)	0.25	$\boxed{0.22}$	0.26	0.30	0.35	0.38	0.40	0.43	0.45	0.47
		(b)	0.27	$\boxed{0.22}$	0.24	0.27	0.30	0.33	0.35	0.36	0.38	0.39
		(c)	0.25	$\boxed{0.22}$	0.27	0.31	0.36	0.36	0.39	0.47	0.56	0.63
36	0.5	(a)	0.62	0.47	0.39	0.35	$\boxed{0.34}$	0.34	0.34	0.35	0.37	0.38
		(b)	0.64	0.49	0.41	0.37	0.35	$\boxed{0.34}$	0.35	0.35	0.36	0.37
		(c)	0.62	0.46	0.38	0.36	$\boxed{0.35}$	0.35	0.36	0.38	0.41	0.42
	1	(a)	0.39	0.26	$\boxed{0.22}$	0.22	0.24	0.25	0.27	0.30	0.32	0.34
		(b)	0.42	0.28	0.24	$\boxed{0.23}$	0.23	0.25	0.26	0.27	0.29	0.30
		(c)	0.39	0.26	$\boxed{0.22}$	0.23	0.24	0.28	0.29	0.32	0.37	0.37
	2	(a)	0.19	$\boxed{0.14}$	0.15	0.18	0.20	0.23	0.25	0.27	0.29	0.31
		(b)	0.21	$\boxed{0.15}$	0.15	0.17	0.18	0.20	0.22	0.23	0.25	0.26
		(c)	0.19	$\boxed{0.14}$	0.16	0.18	0.21	0.25	0.26	0.28	0.34	0.35
18	0.5	(a)	58	67	69	$\boxed{70}$	70	68	68	67	67	64
		(b)	56	65	67	$\boxed{69}$	67	67	66	65	65	64
		(c)	59	67	$\boxed{70}$	69	68	67	63	59	54	48
	1	(a)	71	78	$\boxed{79}$	79	77	75	73	72	71	69
		(b)	71	77	$\boxed{78}$	77	75	74	73	73	71	71
		(c)	71	$\boxed{79}$	79	77	75	74	71	64	58	50
	2	(a)	82	$\boxed{85}$	84	82	80	79	78	76	74	72
		(b)	81	$\boxed{84}$	83	82	81	80	79	79	78	76
		(c)	81	$\boxed{85}$	82	81	78	77	75	69	61	55
36	0.5	(a)	63	72	76	78	79	$\boxed{80}$	79	79	79	77
		(b)	62	71	74	76	78	$\boxed{78}$	78	77	77	77
		(c)	65	72	77	78	78	$\boxed{79}$	78	76	74	74
	1	(a)	77	84	85	$\boxed{86}$	86	86	85	84	82	82
		(b)	76	82	84	$\boxed{86}$	86	85	85	85	84	84
		(c)	77	83	$\boxed{85}$	85	85	84	84	83	80	80
	2	(a)	85	$\boxed{88}$	88	87	86	86	84	84	84	83
		(b)	85	87	$\boxed{88}$	87	87	87	86	85	85	85
		(c)	85	$\boxed{88}$	88	87	86	84	84	82	80	79

Table 4.3: Top panel: the normalized MSEs for the bootstrap variance estimators of $nvar(median(X_1, \cdots, X_n))$ using (a) The grid based block bootstrap (b) The dependent random weighting . The box for each row indicates the smallest normalized MSE among $l = 1, \cdots, 10$. Bottom panel: the empirical coverage (in percent) for the bootstrap-based confidence intervals of the median using (a) and (b). The box for each row indicates the best coverage among $l_n = 1, \cdots, 10$ (Nominal level is 95%). 2D-case: $n = 200, 400$, $\lambda_n = 18, 36$ and $\rho = 1$ is fixed.

			l									
λ_n	n		1	2	3	4	5	6	7	8	9	10
18	200	(a)	0.27	0.20	$\boxed{0.18}$	0.19	0.21	0.25	0.26	0.28	0.34	0.36
		(b)	0.28	0.21	$\boxed{0.17}$	0.17	0.17	0.18	0.19	0.20	0.21	0.22
	400	(a)	0.39	0.25	0.19	$\boxed{0.18}$	0.20	0.25	0.27	0.31	0.39	0.41
		(b)	0.41	0.27	0.20	0.18	$\boxed{0.17}$	0.17	0.19	0.20	0.21	0.22
36	200	(a)	$\boxed{0.13}$	0.13	0.14	0.15	0.16	0.19	0.20	0.21	0.24	0.25
		(b)	$\boxed{0.13}$	0.13	0.13	0.13	0.14	0.14	0.15	0.15	0.16	0.16
	400	(a)	0.17	0.13	$\boxed{0.12}$	0.12	0.12	0.13	0.15	0.15	0.18	0.18
		(b)	0.17	0.13	0.12	$\boxed{0.11}$	0.11	0.11	0.12	0.12	0.13	0.14
18	200	(a)	81	84	$\boxed{86}$	85	83	81	78	76	70	65
		(b)	80	83	85	$\boxed{86}$	85	86	85	85	85	84
	400	(a)	72	82	$\boxed{85}$	84	83	81	78	74	68	66
		(b)	72	79	83	$\boxed{84}$	84	84	83	84	83	83
36	200	(a)	87	$\boxed{89}$	89	89	88	88	87	87	85	85
		(b)	88	88	88	$\boxed{89}$	89	89	89	89	89	89
	400	(a)	83	86	$\boxed{87}$	87	87	87	87	85	85	84
		(b)	83	85	87	87	88	$\boxed{89}$	89	89	88	88

Table 4.4: Top panel: the normalized MSEs for the bootstrap variance estimators of $nvar(\bar{x}_n)$ using (a) The dependent wild bootstrap (b) The dependent random weighting (c) The grid based block bootstrap. The box for each row indicates the smallest normalized MSE among $l = 1, \cdots, 10$. Bottom panel: the empirical coverage (in percent) for the bootstrap-based confidence intervals of the mean using (a), (b) and (c). The box for each row indicates the best coverage among $l_n = 1, \cdots, 10$ (Nominal level is 95%). 2D-case: $n = 200, 400$, $\lambda_n = 18, 36$ and $\rho = 1$ is fixed.

							l					
λ_n	n		1	2	3	4	5	6	7	8	9	10
18	200	(a)	0.36	0.24	0.17	$\boxed{0.15}$	0.15	0.16	0.18	0.19	0.21	0.23
		(b)	0.37	0.25	0.19	$\boxed{0.16}$	0.16	0.16	0.17	0.17	0.18	0.19
		(c)	0.36	0.23	0.17	$\boxed{0.16}$	0.17	0.21	0.23	0.26	0.35	0.38
	400	(a)	0.49	0.31	0.22	0.18	$\boxed{0.17}$	0.18	0.19	0.21	0.23	0.25
		(b)	0.51	0.33	0.24	0.20	$\boxed{0.18}$	0.18	0.19	0.20	0.21	0.22
		(c)	0.49	0.30	0.20	$\boxed{0.19}$	0.19	0.23	0.25	0.30	0.38	0.40
36	200	(a)	0.12	0.080	0.061	0.055	$\boxed{0.054}$	0.057	0.064	0.071	0.080	0.087
		(b)	0.12	0.085	0.065	0.057	$\boxed{0.055}$	0.056	0.060	0.064	0.071	0.077
		(c)	0.11	0.079	0.061	$\boxed{0.055}$	0.058	0.064	0.073	0.084	0.11	0.11
	400	(a)	0.22	0.14	0.094	0.073	$\boxed{0.064}$	0.064	0.067	0.074	0.083	0.095
		(b)	0.23	0.15	0.10	0.080	0.069	$\boxed{0.066}$	0.067	0.071	0.077	0.084
		(c)	0.22	0.14	0.092	0.072	$\boxed{0.068}$	0.070	0.079	0.086	0.11	0.12
18	200	(a)	79	84	87	88	88	88	$\boxed{89}$	88	87	86
		(b)	78	84	86	$\boxed{88}$	87	88	88	88	88	87
		(c)	78	85	$\boxed{88}$	87	86	83	80	77	71	67
	400	(a)	69	80	85	87	$\boxed{88}$	87	87	86	86	85
		(b)	69	79	84	85	$\boxed{87}$	87	87	87	86	86
		(c)	70	82	$\boxed{85}$	85	84	81	78	74	70	66
36	200	(a)	89	91	91	92	$\boxed{93}$	92	92	92	92	92
		(b)	89	90	$\boxed{92}$	92	92	92	92	92	92	92
		(c)	89	91	91	$\boxed{92}$	92	90	90	89	88	87
	400	(a)	83	87	89	90	90	$\boxed{91}$	91	91	91	91
		(b)	83	87	89	89	$\boxed{91}$	90	91	91	91	91
		(c)	83	87	89	$\boxed{90}$	89	89	88	87	86	85

CHAPTER 5

A SUBSAMPLED DOUBLE BOOTSTRAP FOR MASSIVE DATA

5.1 Introduction

In the past decade, we have witnessed massive data (or big data) generated in many fields. Datasets grow in size in part because they are increasingly being collected by ubiquitous information-sensing mobile devices, remote sensing technologies, and wireless sensor networks, among others. Although our computing power has also been advancing steadily, the surge of massive data presents challenges to both computer scientists and statisticians in terms of data storage, computation and statistical analysis. As nicely summarized in Jordan (2013), a key question for statistical inference in the massive data context is "Can you guarantee a certain level of inferential accuracy within a certain time budget even as the data grow in size"? From a statistical point of view, there is a great need for new methods that are theoretically sound and remain computationally feasible even for massive data sets. The classical theoretical criteria to assess the quality of an inferential procedure such as mean squared error, size/power are still relevant, but for massive data, computational efficiency and algorithm quality are also important considerations in comparing different statistical methods and procedures.

With any statistical inference method, an inextricably associated problem is to assess the precision of that inference, and this remains important for the statistical analysis of massive data sets. For example after parameter estimation from a data set, a natural next step is to measure how precise the estimation method is, and this can be measured by the mean squared error, width of confidence interval, and so on. The bootstrap (Efron (1979)) is a powerful and popular procedure that can be applied to estimate precision for a wide variety of inference methods. It has well-known statistical properties including consistency and higher-order accuracy under quite general settings. It is conceptually appealing as it is straightforward to implement, using resamples from the data as a proxy for samples, and is automatic in nature such that the user can implement it without advanced statistical knowledge. However, the benefits of bootstrap come at a considerable computational cost. Each iteration of the bootstrap involves the calculation of a statistical function on a resample of the original data. For a data of size n, on average each resample includes $0.63n$ distinct sample points — therefore each iteration of the bootstrap carries a computational cost of the same order as that of the original inference on the data. Even though this problem can be alleviated with the advent of modern parallel computing platforms, it is still quite overwhelming to repeatedly process such resampled datasets for data of huge size, say a terabyte. Therefore, this calls for new bootstrap methods that are computationally scalable while maintaining good statistical properties.

In their recent work Kleiner et al. (2014) introduced a new resampling procedure called Bag of Little Bootstraps (BLB, hereafter). This procedure consists of randomly selecting small subsets of the data, and then performing a bootstrap on each subset, by constructing weighted resamples of the subset such that the resample size equals the size of the original data. The estimator is calculated on these resamples in the same manner as bootstrap. It is worth noting that this method bears some resemblance to the traditional subsampling (Politis and Romano (1994a)) or m out of n bootstrap (Bickel et al. (1997)), which involve subsamples or resamples of size much smaller than the bootstrap, thereby reducing the computational cost. However, these methods (subsampling or m out of n bootstrap) usually require a rescaling of the output, to adjust for the difference between sample size and resample or subsample size. This feature makes them less user-friendly, since in order to evaluate the precision of an estimation method, the practitioner typically needs to know the rate of convergence of the estimator being used. Additionally, as demonstrated in Kleiner et al. (2014), the performance of subsampling or m out of n bootstrap depends quite strongly on the choice of parameters such as subsample size. By contrast, the resamples in BLB are of the same size as the data, so no rescaling of output is needed thereby retaining the automatic and user-friendly nature of the bootstrap. On the other hand, although the resamples are nominally of the same size as the original data, they contain only a small number of distinct points coming from the subset, which reduces the computational cost of calculating a statistical function of the resamples. The estimates of precision from a few subsets can be averaged to obtain the BLB estimate of precision.

In Kleiner et al. (2014) the authors recommend a large number of resamples from each subset, and a small number of random subsets. However, this means that only a small fraction of the original data is used in computing the BLB estimate, as a large majority of data points may not appear in any of the subsets used. Additionally, running a large number of resamples on each subset might incur high computational costs, even if each resample has less runtime than bootstrap resamples. These two issues can affect the performance of BLB in terms of statistical accuracy and computational cost, respectively.

When facing the trade-off between statistical accuracy and computational cost, a practical question we need to answer is: “given a certain computation time budget, how can a practitioner optimally use that budget to come up with an estimate of precision?” The bootstrap has an obvious answer to this question — keep taking resamples until the budget runs out. This answer holds true irrespective of the statistical inferential method whose precision is of interest. However, it is not obvious how to answer this question for BLB, since it is not clear how to optimize the two tuning parameters — namely number of resamples per subset and the number of subsets, under the time budget constraint. Two natural strategies would be — with a fixed number of resamples per subset use as many random subsets as possible, or with a fixed number of random subsets use as many resamples per subset as possible. Both strategies might be sub-optimal in practice, depending upon the particular problem at hand. Kleiner et al. (2014) suggest a novel adaptive method for selecting the tuning parameters, where one first fixes a tolerance parameter, and then for each subset, one can keep taking resamples till that tolerance level is reached. This method provides a nice way of adaptively choosing tuning parameters for a given level of desired accuracy. However for a given computational time budget, the variability of the precision estimate is not known a priori, and hence it is not clear how to choose an appropriate value of the tolerance parameter that is neither too ambitious nor too conservative for the inference method of interest.

In this chapter we present a new resampling procedure called the Subsampled Double Bootstrap (SDB, hereafter) for massive data. Double bootstrap was first proposed by Beran (1988) as a way of improving the accuracy of bootstrap, but is considerably more expensive than bootstrap and becomes computationally infeasible for massive data. Fast double bootstrap (FDB, hereafter), which was independently proposed by White (2000) and Davidson and MacKinnon (2000,2007), is an interesting alternative that only resamples once in the second stage of bootstrapping and can dramatically speed up the double bootstrap. The FDB has been applied to many tests in econometrics, see Davidson and MacKinnon (2002), Ahlgren and Antell (2008), Richard (2009), among others. Recently, Giacomini et al. (2013) applied the idea of FDB to reduce the computational cost in running

Monte carlo experiments to assess the performance of bootstrap estimators and tests. They demonstrated the consistency of this method and called it a 'warp-speed method' to emphasize its rapidness. Chang and Hall (2014) recently studied the higher order accuracy of FDB in terms of bias correction and coverage accuracy of confidence intervals. In the massive data context, the FDB is still too expensive since its computational cost is about twice the cost of bootstrap. Therefore we propose to do subsampling first and then apply the idea of a single resample in the double bootstrap step to the randomly drawn subset of massive data, to evaluate the precision of a statistical inference method. Since our method is a combination of subsampling and double bootstrap, we call it subsampled double bootstrap (SDB). In the implementation of SDB, we randomly draw a large number of small subsets of the data, but instead of bootstrapping the subsets we construct only one resample from each subset. Since these resamples have the same nominal size as the original data but only a small number of distinct points, SDB retains the automatic nature and computational strength of BLB. The ensemble of resamples is then used to estimate the precision of the inference method, in the same manner as bootstrap. Note that SDB inherits certain features from FDB but is computationally much cheaper than FDB. The number of distinct points in the first-stage subsample and the second-stage resample of SDB are small compared to the number of distinct points in the first-stage and second-stage resamples of FDB, and this makes SDB much faster.

To see the statistical and computational advantages of SDB, note that the estimation time of one SDB iteration is comparable to that of two resamples for a BLB subset, and hence SDB can cover a large number of random subsets in the time it takes BLB to complete a large number of resamples for a single random subset. Hence, SDB can provide a much more comprehensive coverage of the data than BLB within a given time budget. Further, given a certain computational time budget, utilizing that budget with SDB is straightforward as it does not require the choice of any tuning parameters. The practitioner can, just like bootstrap, simply keep running subset-resamples until the time budget runs out.

The rest of the chapter is organized as follows. In Section 6.1.1 we describe SDB in independent data setting. Section 5.3 demonstrates the consistency of SDB for independent data, and Section 5.4 reports two simulation studies comparing SDB, BLB, and bootstrap for independent data. We introduce a time series version of SDB in Section 5.5. Section 5.6 demonstrates consistency for the dependent case, and Section 5.7 reports two simulation studies for time series data. We provide a data illustration on a large meteorological time series dataset in Section 5.8, and the chapter concludes with discussion in Section 5.9. Proofs of the theoretical results and some supplementary simulation results are at the end.

5.2 SDB for independent data

Consider an i.i.d. sample $\mathcal{X}_n = \{X_1, \ldots, X_n\}$ drawn from some unknown distribution P. The parameter of interest is $\theta = \theta(P)$, for which an estimate $\hat{\theta}_n = \hat{\theta}(\mathcal{X}_n)$ is obtained from the sample. (Please see the discussion following Theorem 5.3.1 for a more rigorous definition of the types of parameter and estimator covered under the scope of SDB.) Having chosen the estimator, the statistician often seeks to obtain further information regarding the precision of the estimator $\hat{\theta}(\mathcal{X}_n)$. This requires the estimation of some measure involving the sampling distribution of $\hat{\theta}(\mathcal{X}_n)$ and the true value of the parameter θ. For example, the precision of an estimator can be measured by the mean squared error or the width of a 95% confidence interval for θ.

Such measures of precision can usually be defined in terms of a root function $T_n(\hat{\theta}_n, \theta)$ involving the estimator and the parameter. Let $Q_n = Q_n(P)$ be the unknown sampling distribution of T_n, and assume that the precision measure can be represented as $\xi(Q_n)$ for a suitable functional $\xi(\cdot)$. For example, suppose θ is the population mean, $\hat{\theta}(\mathcal{X}_n)$ is the sample mean, and the measure of interest is the scaled MSE $nE[(\hat{\theta}_n - \theta)^2]$. In our notation, we define the root as $T_n(\hat{\theta}_n, \theta) = \sqrt{n}(\hat{\theta}_n - \theta)$, and define the functional as $\xi(Q_n) = \int x^2 dQ_n(x)$, where Q_n is the sampling distribution of $T_n(\hat{\theta}_n, \theta)$.

Estimation of $\xi(Q_n)$ can be performed by a resampling method like bootstrap. Let $\mathbb{P}_n$ be the empirical distribution of the sample $\mathcal{X}_n$, then we can approximate $Q_n(P)$ by $\hat{Q}_n = Q_n(\mathbb{P}_n)$. To do so, we generate a large number (R) of resamples $\mathcal{X}_n^{*j} = \{X_{j_1}, \ldots, X_{j_n}\}, j = 1, \ldots, R$ from the observed sample $\mathcal{X}_n$. Treating the original estimate $\hat{\theta}(\mathcal{X}_n)$ as the population parameter and the resample estimate $\hat{\theta}(\mathcal{X}_n^{*j})$ as an estimated value of this parameter, we compute the root $T_n(\hat{\theta}(\mathcal{X}_n^{*j}), \hat{\theta}(\mathcal{X}_n))$ for each resample, and obtain the empirical distribution $\hat{Q}_{n,R}$ of this ensemble of roots. Conditionally on $\mathcal{X}_n$, the empirical distribution $\hat{Q}_{n,R}$ converges to the resampling distribution $Q_n(\mathbb{P}_n)$ as R goes to infinity. The underlying idea of the bootstrap is to estimate the unknown sampling distribution Q_n of the root function by this empirical distribution $\hat{Q}_{n,R}$, and estimate the measure $\xi(Q_n)$ by the plug-in estimator $\xi(\hat{Q}_{n,R})$.

In conventional bootstrap, each resample contains an average of $0.63n$ distinct sample points — the computational cost of calculating each resample estimate $\hat{\theta}(\mathcal{X}_n^*)$ is therefore comparable to those of the original sample. Running R iterations requires performing this task R times, which can be computationally demanding for massive datasets, particularly when it involves computation of complex statistics. This limits the application of bootstrap for massive datasets.

For BLB, we fix a subset size b (typically $b = n^\gamma$ for some $0 < \gamma < 1$) and construct a suitable number (S) of random subsets, $\mathcal{X}_{n,b}^{*j} = \{X_{j_1}, \ldots, X_{j_b}\}, j = 1, \ldots, S$, from the observed sample $\mathcal{X}_n$. Next, for each subset $\mathcal{X}_{n,b}^{*j}$, we generate R weighted resamples of size n — this can be represented by $(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,k)}), k = 1, \ldots, R$ where $\mathcal{W}_{n,b}^{*(j,k)} = \{W_1, \cdots, W_b\}$ is a vector representing the frequencies of $(\mathcal{X}_{n,b}^{*j})$ in the k^{th} resample. The weight vector $\mathcal{W}_{n,b}^{*(j,k)}$ is generated from a multinomial distribution with n trials and b cells with uniform chance for each cell, independently of the subset. Treating $\hat{\theta}(\mathcal{X}_{n,b}^{*j})$ as the population parameter and the resample estimate $\hat{\theta}(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,k)})$ as an estimated value of this parameter, we compute the root $T_n(\hat{\theta}(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,k)}), \hat{\theta}(\mathcal{X}_{n,b}^{*j}))$ for each resample, and obtain the empirical distribution $\hat{Q}_{n,b,R}^j$ of this ensemble of roots. In the same spirit as bootstrap, we apply the plug-in estimator $\xi(\hat{Q}_{n,b,R}^j)$ for each subset, and average over different subsets to obtain the estimator $\frac{1}{S}\sum_{j=1}^S \xi(\hat{Q}_{n,b,R}^j)$ of $\xi(Q_n)$.

We propose a subsampled double bootstrap scheme (SDB) based on subsets in the same manner as BLB, but using only one resample per subset. We fix a subset size b and construct a large number (S) of random subsets, $\mathcal{X}_{n,b}^{*j} = \{X_{j_1}, \ldots, X_{j_b}\}, j = 1, \ldots, S$, from the observed sample $\mathcal{X}_n$. However, we generate only one resample from the j^{th} subset, corresponding to $\mathcal{W}_{n,b}^{*(j,1)}$ as defined above, and calculate a single root $T_n^{*j} = T_n(\hat{\theta}(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)}), \hat{\theta}(\mathcal{X}_{n,b}^{*j}))$ from the resample estimate and the subset estimate. With this ensemble $\{R_n^{*j} : j = 1, \ldots, S\}$, we compute the empirical distribution $\hat{Q}_{n,b,S}$ of roots and estimate $\xi(Q_n)$ using the plug-in estimator $\xi(\hat{Q}_{n,b,S})$. Algorithm 1 outlines the computational steps involved.

Note that the computational advantages of SDB and BLB relative to the bootstrap are applicable only when the estimator $\hat{\theta}$ of interest can take the weighted data representation as its argument. This property holds for a large class of commonly used estimators, including M-estimators. For a subset $\mathcal{X}_{n,b}^*$ with resample weights $\mathcal{W}_{n,b}^*$, the resample estimate can then be expressed as $\hat{\theta}(\mathcal{X}_{n,b}^*, \mathcal{W}_{n,b}^*)$. Since BLB and SDB resamples have nominal size n but only $O(b)$ distinct points, computing the resample estimate for these methods is much cheaper than that for bootstrap, which has $O(n)$ distinct points in the resamples.

```
Input  : Data $\mathcal{X}_n = \{X_1, \ldots, X_n\}$          $\xi(\cdot)$: measure of accuracy
         $\theta$: parameter of interest                      $b$: subset size
         $\hat{\theta}_n$: estimator                          $S$: number of subsets
         $T_n(\hat{\theta}_n, \theta)$: root function

Output: $\xi(\hat{Q}_{n,b,S})$: Estimate of $\xi$
for $j \leftarrow 1$ to $S$ do
    (i) Choose random subset $\mathcal{X}_{n,b}^{*j}$ from $\mathcal{X}_n$
    (ii) Compute $\hat{\theta}(\mathcal{X}_{n,b}^{*j})$ from $\mathcal{X}_{n,b}^{*j}$
    (iii) Generate resample $(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)})$ from $\mathcal{X}_{n,b}^{*j}$
    (iv) Compute resample estimate $\hat{\theta}(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)})$ from $(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)})$
    (v) Compute resample root: $T_n^{*j} = T_n(\hat{\theta}(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)}), \hat{\theta}(\mathcal{X}_{n,b}^{*j}))$
end
1. Compute empirical distribution of roots:
$\hat{Q}_{n,b,S} =$ ecdf of $\{T_n^{*1}, \ldots, T_n^{*S}\}$
2. Calculate the plug-in estimator $\xi(\hat{Q}_{n,b,S})$
```

Algorithm 1: SDB algorithm

5.2.1 Comparison of SDB, BLB, and Bootstrap

For both BLB and SDB, the resample estimation step applies to a resample with $O(b)$ distinct points, whereas in bootstrap the resample has $O(n)$ distinct points. This makes SDB and BLB computationally much cheaper than bootstrap when $b << n$. Denote the computational time for performing the estimation process $\hat{\theta}$ on a sample of size m by $t(m)$. In this formulation of computational time we focus on sample size to illustrate the resampling methods, and ignore other factors affecting computational time. For an estimator that can take the weighted data representation, the estimation time for a resample with nominal size n but only b distinct points is $t(b)$. The estimation time for bootstrap, BLB, and SDB, for conducting inference in one original data sample, are listed in Table 5.1, where the symbols have the same meaning as earlier. Bootstrap requires estimation on the original data and its R resamples. Each BLB subset requires estimation on the subset and its R resamples. Each SDB subset requires estimation on the subset and the single resample.

For BLB, Kleiner et al. (2014) recommends R = 100 and a small value of S (2-10 depending on b). For illustration, let $n = 100,000$ and $b = n^{0.6}$, then the number of distinct points in each resample is at most 1000, resulting in much faster computation than bootstrap. However, in terms of sample coverage, each subset

Name	Estimation Time
Bootstrap	$(R+1) \times t(n)$
BLB	$S(R+1) \times t(b)$
SDB	$2S \times t(b)$

Table 5.1: Estimation time for different resampling methods

can cover at most 1% of the data, so 10 subsets can at best cover 10% of the data at an expense of 1010×$t(b)$. The SDB can run more than 500 different subsets at the same expense, providing a far more comprehensive coverage of the data.

Further, given a certain time budget, it is not clear how to choose the tuning parameters R and S that will provide optimal statistical accuracy for BLB. The adaptive method proposed by Kleiner et al. (2014) provides an interesting alternative by choosing a tolerance parameter ϵ instead of R and S. But even then, it is not clear how to choose an appropriate ϵ in practice, since $t(m)$ is not known a priori, and neither do we know the estimation variability as a function of sample size. For the SDB (with a given subset size) and the bootstrap, the estimation time involves only one parameter, the number of resamples (or subsets), and hence the practitioner can simply keep running resamples until the time budget runs out.

5.3 Theory for independent data

In this section, we provide a theoretical analysis of the SDB in a general empirical process setting. Consider a class of functions $\mathcal{F}$ [each element mapping from $\mathbb{R}^k$ to $\mathbb{R}$]. Denote by $\ell^\infty(\mathcal{F})$ the space of bounded functions which map from $\mathcal{F}$ to $\mathbb{R}$. To describe consistency of the SDB, consider the *SDB-process*

$$\hat{\mathbb{G}}_{n,b}^{B}(f) := \frac{1}{\sqrt{n}} \sum_{i=1}^{b} (W_{i,n} - n/b) f(X_{R^{-1}(i)}).$$

Here, $W := (W_{1,n}, ..., W_{b,n}) \sim \text{Multinomial}_b(n, 1/b, ..., 1/b)$ independent of $X_1, ..., X_n$ and R follows a uniform distribution on the permutations of $\{1, ..., n\}$ and is independent of $X_1, ..., X_n, W$. Note that in empirical process settings, it is important to specify the underlying probability space. This is done in the mathematical appendix [see Section 5.10.1]. In order to show that the SDB 'works' in a process setting, we need to establish that the distribution of the SDB process $\hat{\mathbb{G}}_{n,b}^{B}$ [conditional on the observations X_i] is close to the distribution of the empirical process

$\mathbb{G}_n$ where

$$\mathbb{G}_n(f) := \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (f(X_i) - \mathbb{E}[f(X_1)])$$

when both are viewed as elements of $\ell^\infty(\mathcal{F})$. To this end we show that the SDB-process converges in distribution, conditionally on the data $X_1, ..., X_n$, to the same Gaussian process as the empirical process $\mathbb{G}_n$.

Theorem 5.3.1 *Assume that $\mathcal{F}$ is a Donsker class for P, that $X_i \sim P$ are i.i.d. and that additionally $\mathcal{F}_\delta := \{f - g : f, g \in \mathcal{F}, P(f - g)^2 \ \leq \delta\}$ is measurable in the sense discussed in Giné and Zinn (1984) for each $\delta > 0$. Then we have for* $\min(n, b) \to \infty$

$$\hat{\mathbb{G}}^B_{n,b} \underset{W,R}{\overset{\mathbb{P}}{\rightsquigarrow}} \mathbb{G}$$

in $\ell^\infty(\mathcal{F})$ where $\mathbb{G}$ denotes a centered Gaussian process with covariance $\mathbb{E}[\mathbb{G}(f)\mathbb{G}(g)] =$ *cov*$(f(X), g(X))$*.*

In the above Theorem, conditional weak convergence $\underset{W,R}{\overset{\mathbb{P}}{\rightsquigarrow}}$ is in the sense described in Kosorok (2008), Section 2.2.3. A proof of this result can be found in the mathematical appendix [Section 5.10.1].

One remarkable fact about Theorem 5.3.1 is that, in addition to $\mathcal{F}$ being P-Donsker, the only requirement on the class of functions $\mathcal{F}$ is a mild measurability condition. This means that the SDB on a process level 'works' whenever the corresponding functional central limit Theorem holds true (up to the mild measurability assumption on the function class $\mathcal{F}$), i.e. the SDB can be applied in a very wide variety of settings. The proof relies on basic tools from empirical process theory [in particular, a fundamental result on the exchangeable bootstrap, see Theorem 3.6.3 in van der Vaart and Wellner (1996)], but is completely different from the proof of Theorem 1 in Kleiner et al. (2014). The reason is that in the BLB one initial subset is fixed, while in the SDB a different subset of the data is used in each iteration. The latter fact poses additional challenges for the theoretical analysis of SDB.

Theorem 5.3.1 provides a fundamental building block for the analysis of SDB. Combined with the continuous mapping theorem and functional delta method for the bootstrap [see for instance Kosorok (2008), Theorem 10.8 and Theorem 12.1], it can be utilized to validate the consistency of SDB for a wide range of applications. For illustration purposes, let us consider an application of the functional delta method for the bootstrap with the root $T_n(\hat{\theta}_n, \theta) := \sqrt{n}(\hat{\theta}_n - \theta)$. Assume that we are interested in conducting inference on a parameter θ which can be represented as $\phi((f \mapsto Pf)_{f\in\mathcal{F}})$, and the estimator takes the form $\hat{\theta}(\mathcal{X}) = \phi((f \mapsto \mathbb{P}_n f)_{f\in\mathcal{F}})$ for a suitable map ϕ. More precisely, we assume that ϕ satisfies the following condition

(H) There exists a V which is a vector space with $V \subset \ell^\infty(\mathcal{F})$ such that the sample paths of $\mathbb{G}$ lie in V with probability one. The map $\phi : \ell^\infty(\mathcal{F}) \to \mathbb{R}^k$ is compactly differentiable tangentially to V in the point $H : f \mapsto Pf$. Denote the corresponding derivative by ϕ'_H.

For $f \in \mathcal{F}$, write $\mathbb{P}_{n,b} f := \frac{1}{n}\sum_{i=1}^b W_{i,n} f(X_{R^{-1}(i)})$. Then, in the notation from Section 6.1.1, we have $\hat{\theta}(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b}) = \phi((f \mapsto \mathbb{P}_{n,b} f)_{f\in\mathcal{F}})$. Now the delta method for the bootstrap [Theorem 12.1 in Kosorok (2008)] yields for $\min(n, b) \to \infty$

$$T_n(\hat{\theta}(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b}), \hat{\theta}(\mathcal{X}_n)) = \sqrt{n}(\hat{\theta}((\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b})) - \hat{\theta}(\mathcal{X}_n)) \underset{W,R}{\overset{\mathbb{P}}{\rightsquigarrow}} \phi'_H \mathbb{G}.$$

At the same time, the classical functional delta method yields

$$T_n(\hat{\theta}(\mathcal{X}_n), \theta) = \sqrt{n}(\hat{\theta}(\mathcal{X}_n) - \theta) \rightsquigarrow \phi'_H \mathbb{G}.$$

Assume measurability of $\hat{\theta}((\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b})), \hat{\theta}(\mathcal{X}_n)$. Write $\mathcal{L}$ for the distribution of $\phi'_H \mathbb{G}$, $\mathcal{L}_n$ for the distribution of $T_n(\hat{\theta}(\mathcal{X}_n), \theta)$, and denote by $\mathcal{L}^B_{n,b}(R, W)$ the distribution of $T_n(\hat{\theta}(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b}), \hat{\theta}(\mathcal{X}_n))$ conditional on R, W. Denoting by d a metric on the space of distributions on $\mathbb{R}^k$ which metrizes weak convergence, we have proved that $\mathrm{d}(\mathcal{L}_n, \mathcal{L}) \to 0$ as $n \to \infty$ and $\mathrm{d}(\mathcal{L}^B_{n,b}(R, W), \mathcal{L}) \to 0$ in outer probability as $\min(n, b) \to \infty$. In particular, this shows that for any map ξ from the space of distributions to $\mathbb{R}^k$ which is continuous in the point $\mathcal{L}$ with respect to the metric d, we have $\xi(\mathcal{L}^B_{n,b}(R, W)) - \xi(\mathcal{L}) \to 0$ in outer probability. This shows that the conclusion of Theorem 1 in Kleiner et al. (2014) continues to hold in the SDB setting.

5.4 Simulation study for independent data

In this section, we report two simulation studies comparing the performance of bootstrap, BLB, and SDB in large simulated datasets in the i.i.d. framework. We used model settings similar to Kleiner et al. (2014). Since they have already demonstrated that BLB performs better than the m out of n bootstrap and subsampling, we did not include these methods in our study.

5.4.1 Multiple Linear Regression

Consider a d-dimensional multiple linear regression model

$$y_i = \beta_1 x_{i,1} + \ldots + \beta_d x_{i,d} + e_i$$

for $i = 1, \ldots, n$. Our parameter of interest is the d-dimensional vector of slope coefficients, whose true value is $\beta = (\beta_1, ..., \beta_d) = (1, \ldots, 1)'$. We use the usual OLS estimator $\hat{\beta}$. We also want to construct a simultaneous 95% confidence region for β. Traditionally we use the F-statistic

$$T_n(\hat{\beta}, \beta) = \frac{(\hat{\beta} - \beta)' X' X (\hat{\beta} - \beta)/d}{(y - X\hat{\beta})'(y - X\hat{\beta})/(n - d - 1)}$$

to construct the joint confidence region. Let $q_{0.95}$ be the 95% quantile of the true (unknown) distribution of $T_n(\hat{\beta}, \beta)$. Then the confidence region is given by $\{\beta : T_n(\hat{\beta}, \beta) \leq q_{0.95}\}$. In general the true distribution of T_n, and hence its quantile $q_{0.95}$, is unknown. But it can be estimated by the resampling techniques described in the previous section, with $\xi(Q_n) = q_{0.95}$ where Q_n is the true distribution of T_n.

In our simulations, we use a model from the simulation study of Kleiner et al. (2014). We generate $x_{i,j} \overset{iid}{\sim} t_3$ and $e_i \overset{iid}{\sim} N(0, 100)$ independently. For normally distributed errors, we know that $T_n \sim F(d, n-d-1)$, and hence the true quantiles are given by those of the corresponding F distribution. We define the error rate as

$$|\frac{\hat{q}_{0.95}}{q_{0.95}} - 1|$$

where $\hat{q}$ and q represent the estimated and true quantiles of T_n, respectively. We use subset size $b = n^{\gamma}$ with $\gamma = 0.6, 0.7, 0.8$ for both BLB and SDB, and let n=100000, d=100. Following Kleiner et al. (2014) we fix $R = 100$ for BLB. We allowed the competing methods to run for 60 seconds.

5.4.2 Logistic Regression

Consider a d-dimensional multiple logistic regression model

$$y_i \overset{ind}{\sim} Ber(p_i) \text{ where } p_i = \beta_1 x_{i,1} + \ldots + \beta_d x_{i,d}$$

for $i = 1, \ldots, n$. Our parameter of interest is the d-dimensional vector of slope coefficients, whose true value is $\beta = (\beta_1, \ldots, \beta_d) = (1, \ldots, 1)'$. We use the maximum likelihood estimator $\hat{\beta}_n$ which does not have a closed form expression for this model, but can be numerically computed using a Newton-Raphson method. We use the R function *glm* for fitting the model. As before, we want to construct a simultaneous 95% confidence region for β. Define the root function

$$T_n(\hat{\beta}, \beta) = (\hat{\beta} - \beta)'\hat{\Sigma}(\hat{\beta} - \beta)$$

where $\hat{\Sigma} = \sum_{i=1}^{n} \frac{\exp(x_i'\hat{\beta})}{[1+\exp(x_i'\hat{\beta})]^2} x_i x_i'$. Let $q_{0.95}$ be the 95% quantile of the true (unknown) distribution of $T_n(\hat{\beta}, \beta)$. Then the confidence region is given by $\{\beta : T_n(\hat{\beta}, \beta) \leq q_{0.95}\}$. In general the true distribution of T_n, and hence its quantile $q_{0.95}$, is unknown. But it can be estimated by the resampling techniques described in the previous section, with $\xi(Q_n) = q_{0.95}$ as the target precision parameter, where Q_n is the true distribution of T_n.

We generate $x_{i,j} \overset{iid}{\sim} t_3$ and obtain a numerical approximation of $q_{0.95}$ using 10000 Monte Carlo simulations. As before, we define the error rate as $|\hat{q}_{0.95}/q_{0.95} - 1|$. We use subset size $b = n^{\gamma}$ with $\gamma = 0.6, 0.7, 0.8$ for both BLB and SDB, and use n=100000, d=10. Following Kleiner et al. (2014) we fix $R = 100$ for BLB. We allowed the competing methods to run for 20 seconds.

5.4.3 Comparison of SDB, BLB, and Bootstrap

The methods are compared with respect to the time evolution of error rates. Note that this is different from conventional analysis where error rates from competing methods are compared for the same number of iterations. This makes sense because different methods have different estimation time profiles (as formulated in Table 5.1), and we want to investigate which method is the fastest to produce reasonably accurate results. We consider a time grid $1, 2, \ldots, 60$ (in seconds) and at each time point t, we look up the latest iteration that was completed by this time, and calculate the error corresponding to the estimate $\hat{\xi}$ from cumulative iterations including that iteration. For each method and any t, this can be interpreted as the error rate obtained by that method for a given computation time budget of t seconds. Different methods will have different numbers of iterations completed within the same time budget. For bootstrap and SDB, latest iteration means the latest completed resample or subset-resample, while for BLB (following Kleiner et al. (2014)) the latest iteration means the latest completed subset. Note that till the first iteration is complete, we do not have an estimate $\hat{\xi}$, so we consider the error rate to be 1 till the first iteration is completed. Error rates are averaged across 20 Monte Carlo simulations.

Figure 6.1 shows the time evolution of error rates for bootstrap, BLB, and SDB. Bootstrap has the highest computing cost which gets reflected in its slow convergence. The performance of BLB and SDB are close to one another for generous time budgets, but for lower time budgets SDB performs better by quickly giving a reliable estimate while BLB takes some time to complete the first subset. This phenomenon becomes particularly prominent for higher values of b as BLB's computing time for each subset becomes large. For small time budgets even bootstrap can beat BLB when $b = n^{0.8}$, since the time taken by BLB to complete a subset can exceed the given budget. A similar phenomenon for small time budgets was observed in the simulation study of Kleiner et al. (2014) (see Figure 1(a)—(c) in their paper for linear regression and 2 (a)—(c) for logistic regression), where bootstrap estimates are available but BLB estimates are not available yet for subset size $b = n^{0.8}$ or $b = n^{0.9}$.

Remark 5.4.1 Computing time for the resampling methods depends on various aspects of the computational infrastructure used, e.g. the processing power of the computer, storage capacity, and statistical software or computing platform. All our simulations were performed on a desktop computer with Intel(R) Core(TM)2 Duo CPU E8400 @3.00 GHz processor and 4 GB RAM, running R version 3.0.1. The computational infrastructure influences the computing time of various resampling methods in identical manner, so the relative performance of these methods should be qualitatively similar in a different infrastructure, even if the absolute performances might vary.

Remark 5.4.2 It is relevant to note that while we have used models from Kleiner et al. (2014) in these studies, the precision measure and the method of comparison between resampling schemes are slightly different. They constructed marginal confidence intervals for the individual regression coefficients, and combined results for the d coefficients by averaging the error rate over the dimensions. Thus they are interested in measures of precision of the individual estimation tasks of estimating the d coefficients. However, in a multivariate regression setting, the joint estimation task of all coefficients taken together might be of more interest. Accordingly, we constructed a simultaneous confidence region for the d-dimensional vector of regression coefficients to assess precision of this joint estimation task, and compute error in terms of this confidence region.

For comparison between resampling schemes, Kleiner et al. (2014) allowed the competing resampling schemes to converge, and for each iteration (defined as a complete subset for BLB and resample for bootstrap), they computed the average cumulative computing times and average error rates from five Monte Carlo simulations. They compared resampling schemes on the basis of this average time vs average error trade-off. In our simulations, we compare methods on the basis of error rate achieved for a given time budget, over 20 Monte Carlo simulations. Thus time is not averaged across simulations — rather, at a fixed point in time we look up the error rates obtained by this time in the Monte Carlo simulations, and average them. If at a certain time point no estimate is available yet (no iteration has been completed), we assign an error rate of 1.

In particular, in our simulation plots, the error rate changes only when an iteration has been completed. This makes them look 'jerky' and unstable as there are long stretches of a flat line followed by a sudden drop. Since estimates change only upon the completion of a new iteration, the arrival of new estimates is actually an intermittent process rather than a continuous process with respect to the time axis, and the error rate does not change unless a new estimate is available. Therefore it is realistic that error rates change in a 'jerky' fashion rather than smoothly, and this is not a symptom of instability.

Remark 5.4.3 Several bootstrap approaches exist in a regression setting — for example paired bootstrap, residual bootstrap, wild bootstrap and so on. In these simulations we have implemented the paired bootstrap, where both regressors and response are resampled. The paired bootstrap method can be naturally extended to the BLB and SDB algorithms, but it is unclear whether there are straightforward extensions for residual bootstrap or wild bootstrap.

As pointed out by a referee, another alternative is to look at bootstrap p-values instead of confidence regions for regression models. In our formulation ξ is a parameter associated with the sampling distribution Q_n of the root function T_n (which in this case is the F-statistic), while the p-value is a statistic. However, one can implement Algorithm 1 to 'estimate' the true p-value using the empirical distribution $\hat{Q}_{n,b,S}$. Limited (unreported) simulation results suggest that SDB still possesses the same advantage over BLB and bootstrap, which is reported for confidence region. A more careful investigation regarding the suitability of SDB for approximation of the p-value in theory and finite sample simulations is left for future work.

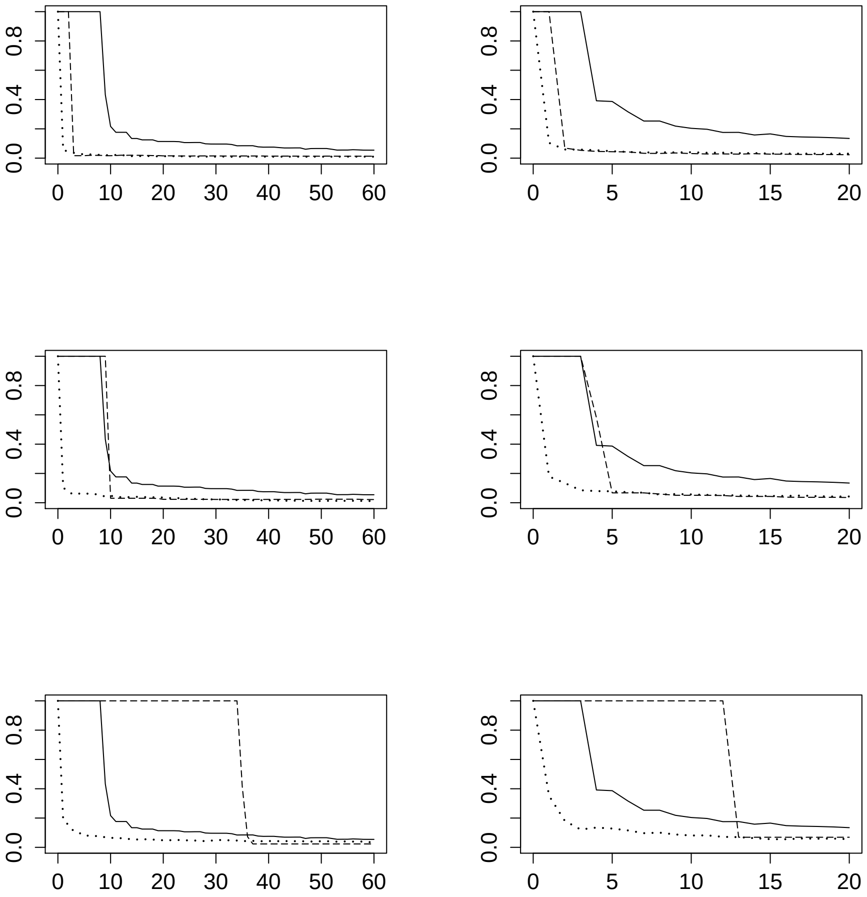


Figure 5.1: Time evolution of error rates for multiple linear regression with d=100 (left column) and multiple logistic regression with d=10 (right column). Sample size n=100000, subset size is $b = n^\gamma$ where $\gamma = 0.6$ (top row), $\gamma = 0.7$ (middle row) and $\gamma = 0.8$ (bottom row). Bootstrap errors are represented by solid lines, BLB errors by dashed lines, and SDB errors by dotted lines. Errors are averaged over 20 simulations.

5.5 SDB for time series data

In this section, we extend SDB to time series data. Note that Kleiner et al. (2014) have briefly mentioned an extension of BLB to the time series setting using stationary bootstrap (Politis and Romano (1994b)), however no rigorous theory is provided. Also see Laptev et al. (2012) for a recent implementation on a large Twitter dataset.

Suppose we observe $\mathcal{X}_n = \{X_t\}_{t=1}^n$, which is a stretch of length n from the strictly stationary time series $\{X_t\}_{t\in\mathbb{Z}}$, and let P denote the joint probability law that governs the stationary sequence. Let $\theta = \theta(P)$ be our parameter of interest, and suppose we have an estimator $\hat{\theta}_n(\mathcal{X}_n)$ which is a measurable mapping from $\mathcal{X}_n$ to $\mathbb{R}$. As with independent data, we are interested in evaluating the precision of this statistical inference. As before, this can be formulated in terms of a root function $T_n(\hat{\theta}_n, \theta)$, and the precision can be expressed as $\xi(Q_n)$ where Q_n is the true (unknown) distribution of T_n.

For BLB, we first construct subsets of the original sample by randomly choosing a continuous stretch of data $\mathcal{X}^*_{n,b} = \{X_{J+i}\}_{i=1}^b$ where $0 \leq J \leq n-b+1$ and the subset size b is fixed beforehand. From the j^{th} subset, we then construct R weighted resamples $(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,k)}_{n,b})$ for $k = 1, \ldots, R$ of size n using the Moving Block Bootstrap (MBB, hereafter) of Künsch (1989) and Liu and Singh (1992). We use MBB instead of stationary bootstrap used by Kleiner et al. (2014) since the MBB is conceptually simpler, and easier in terms of theoretical treatment. For this, we consider some suitable block length $L < b$ and divide the subset $\{X_J, X_{J+1}, \ldots, X_{J+b-1}\}$ into an ensemble of overlapping blocks $\{X_i, X_{i+1}, \ldots, X_{i+L-1}\}$ where $J \leq i \leq J+b-L+1$. The resample is constructed by concatenating blocks that are randomly sampled from this ensemble, till we obtain a chain of size n. Note that when n is not a multiple of L, we will need to take a fraction of the final block in order to obtain a resample of length exactly n. This gives us an ensemble of roots of the form $T_n(\hat{\theta}(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,k)}_{n,b}), \hat{\theta}_b(\mathcal{X}^{*j}_{n,b})), k = 1, \ldots, R$, and we use the empirical distribution of this ensemble to approximate the unknown distribution of $T_n(\hat{\theta}_n, \theta)$. Averaging over $j = 1, \ldots, S$ subset estimates then gives the BLB estimate of precision.

For SDB, for each subset $\mathcal{X}^{*j}_{n,b}$, we generate only one MBB resample $(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b})$ to construct the root $T_n(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b}), \hat{\theta}_b(\mathcal{X}^{*j}_{n,b}))$. We do this a large number (S) of times to generate an ensemble of roots, and use the empirical distribution of this ensemble to approximate the unknown distribution of $T_n(\hat{\theta}_n, \theta)$. Algorithm 2 outlines the computational steps involved.

In the time series case, estimation time can be formulated as $t(m)$ in a manner

```
Input  : Data $\mathcal{X}_n = \{X_1, \ldots, X_n\}$          $\xi(\cdot)$: measure of accuracy
         $\theta$: parameter of interest                    $b$: subset size
         $\hat{\theta}_n$: estimator                         $L$: block length
         $T_n(\hat{\theta}_n, \theta)$: root function        $S$: number of subsets

Output: $\xi(\hat{Q}_{n,b,S})$: Estimate of $\xi$
for $j \leftarrow 1$ to $S$ do
    (i) Choose random subset $\mathcal{X}^{*j}_{n,b} = \{X_{J+i}\}_{i=1}^{b}$ from $\mathcal{X}_n$ where
    $0 \leq J \leq n - b + 1$
    (ii) Compute $\hat{\theta}(\mathcal{X}^{*j}_{n,b})$ from $\mathcal{X}^{*j}_{n,b}$
    (iii) Choose $k = n/L$ blocks by randomly sampling $k$ starting
    points $(t_1, \ldots, t_k)$ from $\{J + 1, \ldots, J + b - L + 1\}$ with replacement
    (iv) Construct resample weights for subset:
    Initialize: $\mathcal{W}^{*(j,1)}_{n,b} \leftarrow (\underbrace{0 \ldots 0}_{b})$
    for $i \leftarrow 1$ to $k$ do
        $\mathcal{W}^{*(j,1)}_{n,b} \leftarrow \mathcal{W}^{*(j,1)}_{n,b} + (\underbrace{0 \ldots 0}_{t_i - 1}\ \underbrace{1 \ldots 1}_{L}\ \underbrace{0 \ldots 0}_{b - t_i - L + 1})$
    end
    (v) Compute resample estimate $\hat{\theta}(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b})$ from $(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b})$
    (vi) Compute resample root: $R^{*j}_n = T_n(\hat{\theta}(\mathcal{X}^{*j}_{n,b}, \mathcal{W}^{*(j,1)}_{n,b}), \hat{\theta}(\mathcal{X}^{*j}_{n,b}))$
end
1. Compute empirical distribution of roots:
$\hat{Q}_{n,b,S} =$ ecdf of $\{R^{*1}_n, \ldots, R^{*S}_n\}$
2. Calculate the plug-in estimator $\xi(\hat{Q}_{n,b,S})$
```

Algorithm 2: SDB time series algorithm

similar to Section 5.2.1, where m is the number of distinct points, and the estimation times listed in Table 5.1 apply with MBB taking the place of bootstrap. MBB requires estimation on the original data and its R resamples. Each BLB subset requires estimation on the subset and its R resamples. Each SDB subset requires estimation on the subset and the single resample.

Broadly speaking, the time series version of SDB retains the advantages discussed in Section 5.2.1 in the context of independent data. For a given computational time budget, by using a single resample for each random subset SDB can accommodate much more comprehensive coverage of data than BLB. Also, BLB involves tuning parameters R and S (or ϵ under the adaptive method) whose selection can be non-trivial, while SDB and MBB do not require this type of tuning parameter selection. However, an important tuning parameter in the time series setting is the block length L which can affect both variability and the accuracy of the estimate of precision in all three resampling methods.

5.6 Theory for dependent data

We begin by setting up a mathematical framework for SDB in the dependent case. Throughout this section we assume that the observations $X_1, ..., X_n$ stem from a strictly stationary time series $\{X_t\}_{t \in \mathbb{Z}}$. Given a sample $X_1, ..., X_n$, the SDB procedure for time series can be described through the following steps.

1. Pick a random variable J which is distributed uniformly on $0, ..., n-b-1$. This corresponds to the first step of randomly selecting a block of length b from the complete data.

2. Choose $K = \lceil n/L \rceil$ random variables $s_1, ..., s_K$ which are i.i.d. and distributed uniformly on $0, ..., b-L+1$. Generate the sample $X_1^*, ..., X_n^*$ by setting
$$(X^*_{kL+1}, ..., X^*_{(k+1)L}) := (X_{J+s_k}, ..., X_{J+s_k+L-1})$$

3. After the first two steps above, one realization of the SDB process is given by
$$\hat{\mathbb{G}}^B_{n,b}(f) := \frac{1}{\sqrt{n}} \sum_{i=1}^n \Big(f(X_i^*) - \frac{1}{b} \sum_{j=1}^b f(X_{J+j}) \Big).$$
Repeat a large number of times, each time generating a new $J, s_1, ..., s_K$.

As in the case of independent observations, our aim is to establish validity of the SDB for general classes of functions. In contrast to the i.i.d. setting, where the 'classical' bootstrap for empirical processes is well understood, there are very few results on bootstrap validity for general empirical processes. All of the available results rely on the notion of β-mixing to measure dependence. More precisely, the k'th β-mixing coefficient $\beta(k)$ with $k \in \mathbb{N}$ is defined as

$$\beta(k) := \frac{1}{2} \sup \sum_{(i,j) \in I \times J} |P(A_i \cap B_j) - P(A_i)P(B_j)|$$

where the supremum is taken over all finite measurable partitions $(A_i)_{i \in I}, (B_j)_{j \in J}$ of $\sigma(X_t, t \leq 0)$ and $\sigma(X_t, t \geq k)$, respectively. As of this writing, we are aware of only three articles that deal with bootstrap validity for general empirical processes based on dependent observations. Bühlmann (1995) considers the moving blocks bootstrap under exponential decay of the β-mixing coefficients and classes of functions with polynomial bracketing numbers. Radulović (1996) establishes the validity of the moving blocks bootstrap for VC [see van der Vaart and Wellner (1996), Chapter 2.6.2 for a definition] classes of functions under conditions on polynomial decay of β-mixing coefficients. Finally, Radulović (2009) revisits the disjoint blocks bootstrap and proves its validity under generic conditions on the function class and decay of β-mixing coefficients. The latter paper also contains a nice overview of literature on bootstrap validity under dependence [see also Radulović (2002) for a review of earlier results]. Our main result can be viewed as an analogue of Theorem 1 in Radulović (1996).

Theorem 5.6.1 *Assume that $\mathcal{F}$ is a permissible [as defined on page 228-229 in Kosorok (2008)] VC class with envelope function F such that $\mathbb{E}[F^p(X_1)] < \infty$ for some $p > 2$. Assume that the mixing coefficients β satisfy $\beta(k) \leq k^{-q}$ for some $q > p/(p-2)$. If additionally there exist $\kappa > 0, \gamma > 0, 0 < \rho < \frac{p-2}{2(p-1)}$ such that b, L satisfy*

$$n^{-1/2} L b^{\gamma} = o(n^{-\kappa}), \quad L \to \infty, \quad L = O(b^{\rho}), \quad n^{1/2} = O(b^{(p-1)\gamma}),$$

we have

$$\hat{\mathbb{G}}^{B}_{n,b} \underset{J,S}{\overset{\mathbb{P}}{\rightsquigarrow}} \mathbb{G}$$

in $\ell^{\infty}(\mathcal{F})$ where $\mathbb{G}$ denotes a centered Gaussian process with covariance structure

$$\mathbb{E}[\mathbb{G}(f)\mathbb{G}(g)] = \sum_{t \in \mathbb{Z}} (\mathbb{E}[f(X_1)g(X_t)] - \mathbb{E}[f(X_1)]\mathbb{E}[g(X_1)]).$$

A proof of this theorem is in the mathematical appendix [Section 5.10.2]. Theorem 5.6.1 shows that the time series version of SDB also works in a wide range of settings. In particular, the continuous mapping theorem and delta method for the bootstrap can be employed in the same fashion as discussed at the end of Section 5.3. We conjecture that the assumptions on the dependence can be weakened if we consider more specialized classes of functions, such as indicators of rectangles which would lead to the 'classical' empirical distribution function.

5.7 Simulation study for time series

In this section we report the numerical performance of SDB, BLB, and MBB in two simulation studies involving large time series data.

5.7.1 Median of AR(1) process

Consider an AR(1) time series formulated as

$$X_t = \rho X_{t-1} + e_t$$

of length $n = 100,000$ and random innovation $e_t \overset{iid}{\sim} N(0,1)$. The parameter of interest is the population median M. We define $T_n = \sqrt{n}(M_n - M)$ where M_n is the sample median. We are interested in evaluating the precision of the estimator M_n. Our measure of precision is a quantile of the distribution of T_n, i.e. $\xi = q_\alpha(T_n)$ which can be used for constructing confidence intervals, for example, with $\alpha = 5\%, 95\%$ we can construct a 90% confidence interval.

We obtain the 'true' value ξ_{true} from 10000 simulations. The error rate is measured by $\mid \hat{\xi}/\xi_{true} - 1 \mid$. We implement and compare the three resampling methods, namely MBB, BLB, and SDB. Block length is $L = 10, 20, 50$ for all methods, and we use subset sizes $b = 5000, 10000$ for BLB and SDB. We allow each method to run for 60 seconds for $L = 20, 50$ and 120 seconds for $L = 10$, to allow BLB to complete one subset.

5.7.2 Time Series Regression

We also studied the relative performance of MBB, BLB, and SDB in the time series regression framework (see e.g. Andrews and Monahan (1992), Kiefer et al. (2000), Rho and Shao (2013)). Consider the time series regression model

$$y_t = X_t'\beta + u_t$$

for $t = 1, \ldots, n$ where β is a $d \times 1$ vector of regression coefficients, X_t is a $d \times 1$ vector of stationary regressors, and u_t is a stationary error process that satisfies $\mathbb{E}[u_t \mid X_t] = 0$. We considered the AR(1)-HOMO regression model of Andrews and Monahan (1992) where the d regressors and the error process are mutually independent, mean zero, homoskedastic, AR(1) processes with autocorrelation ρ and standard normally distributed innovations, and set $\beta = \mathbf{0}$. We set $n = 100,000$ and $d = 10$, and use $\rho = -0.8, 0.5, 0.9$. Similar to Section 5.4.1, the parameter of interest is β, estimator of choice is the least-squares estimate $\hat{\beta}$, and we measure precision by constructing a 95% confidence region for β using the F-statistic. We obtain the 'true' value of $q_{0.95}$ from 10000 simulations. We define error rate by $|\hat{q}_{0.95}/q_{0.95} - 1|$ as before, and use block lengths $L = 10, 20, 50$, subset sizes $b = 5000, 10000$ for

BLB and SDB. To allow BLB to complete one subset, we ran each method for 150 seconds for $L = 10$, 90 seconds for $L = 20$, and 60 seconds for $L = 50$.

5.7.3 Comparison of SDB, BLB, and MBB

The methods are compared with respect to the time evolution of error rates, as discussed in Section 5.4.3. Error rates for different methods are averaged across 20 Monte Carlo simulations. Results for $L = 50$ are displayed in Figures 5.2, 5.3, and 5.4. To save space, results for $L = 10, 20$ are presented at the end of theoretical proofs. We can see that SDB shows significant advantages over its competitors. In particular, it is encouraging to observe that for shorter time budgets (half of total runtime or less), SDB has a clear advantage over the other methods in most cases. SDB can give a reasonable estimate by 10-15 seconds in most cases, while the BLB can take a substantial time to complete a single subset. MBB has highest computing cost which is reflected in its slow convergence, but it appears that MBB can often provide a reasonable estimate by the time taken by BLB to complete one subset, which is consistent with our finding in the iid case.

An interesting aspect of these results is that block length affects both running time and accuracy. The behavior of the resample estimate depends on block length, and this affects accuracy of the resampling methods. The dependence of running time on block length comes from the fact that construction of resample weights (Step (iv) of Algorithm 2) depends on the number of blocks in the resamples, and this step affects running time of the algorithms. However, the advantages of SDB in our numerical results are consistent over the various values of block length used.

5.8 Data Analysis

We apply our method to analyze the Central England Temperature (CET) dataset, which is a meteorological time series dataset consisting of 228 years (1780-2007) of average daily temperatures in central England. The CET dataset represents the longest continuous thermometer-based temperature record on earth, and was previously analyzed by Zhang et al. (2011) and Berkes et al. (2009) in the context of inference for functional time series. In our analysis, we treat the dataset as an univariate time series sample of daily average temperatures. The sample size is $n = 228 \times 365 = 83220$, where we ignore leap years. We remove seasonality by subtracting from each observation the mean temperature for that calendar day

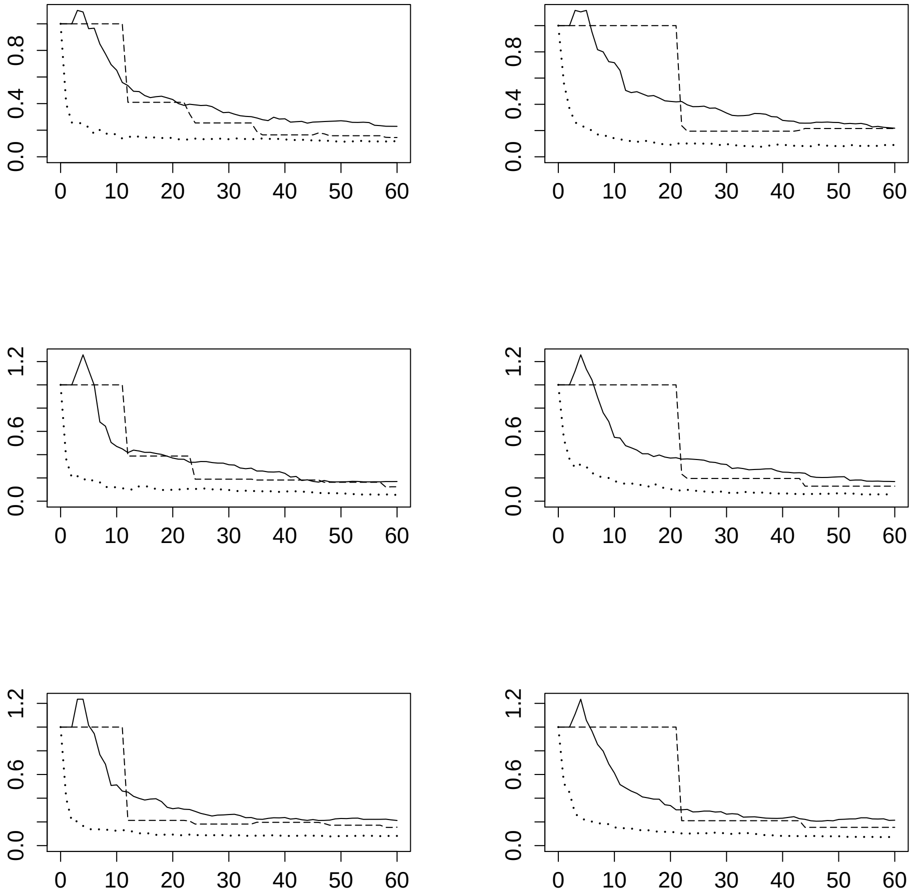


Figure 5.2: AR(1) simulation results with ξ = 95% quantile of $T_n = \sqrt{n}(M_n - M)$, sample size n=100000, block length L=50, autocorrelation ρ = -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size b = 5000 (left column) ,10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 120 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

across 228 years. Our parameter of interest is the population mean μ of this univariate time series. We use sample mean $\bar{X}$ (calculated from the $n = 83,220$ observations after removing seasonality) as the estimator of μ, and we want to construct a 90% confidence interval for μ to assess the quality of estimation. We define $T_n = \sqrt{n}(\bar{X} - \mu)$ as the root function, and let the precision measure $\xi = (q_{0.95} - q_{0.05})$ be the width of the 90% confidence interval.

We applied MBB, BLB, and SDB on this dataset with block length $L = 10, 20, 50$

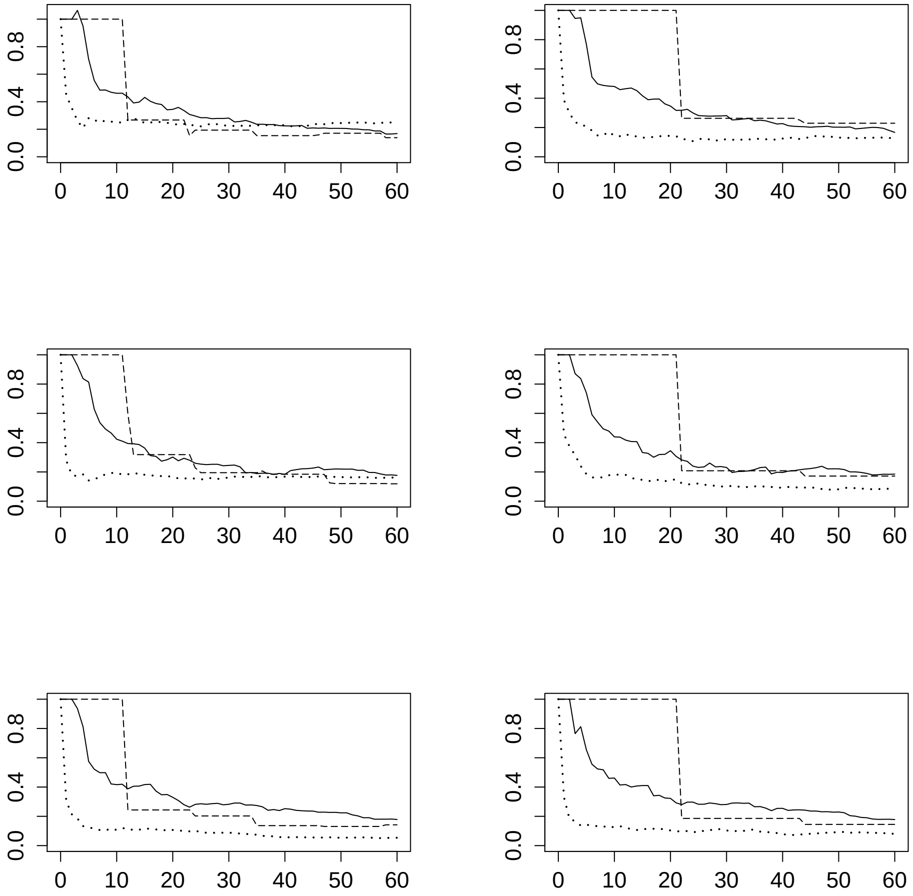


Figure 5.3: AR(1) simulation results with $\xi = 5\%$ quantile of $T_n = \sqrt{n}(M_n - M)$, sample size n=100000, block length L=50, autocorrelation $\rho =$ -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size $b = 5000$ (left column) ,10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 120 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

and subset size $b = 5000, 10000$. MBB was allowed to run for 600 seconds, while BLB and SDB were allowed to run for 300 seconds. Figure 5.5 displays the time evolution of $\hat{\xi}$ for the competing methods. Note that in this empirical example the true width is not known, however it appears that for any given block length, the three methods converge to similar estimates of the width. MBB is the slowest to converge, and continues to display substantial oscillations well after BLB and SDB have stabilized. BLB and SDB quickly converge to stable estimates, but for

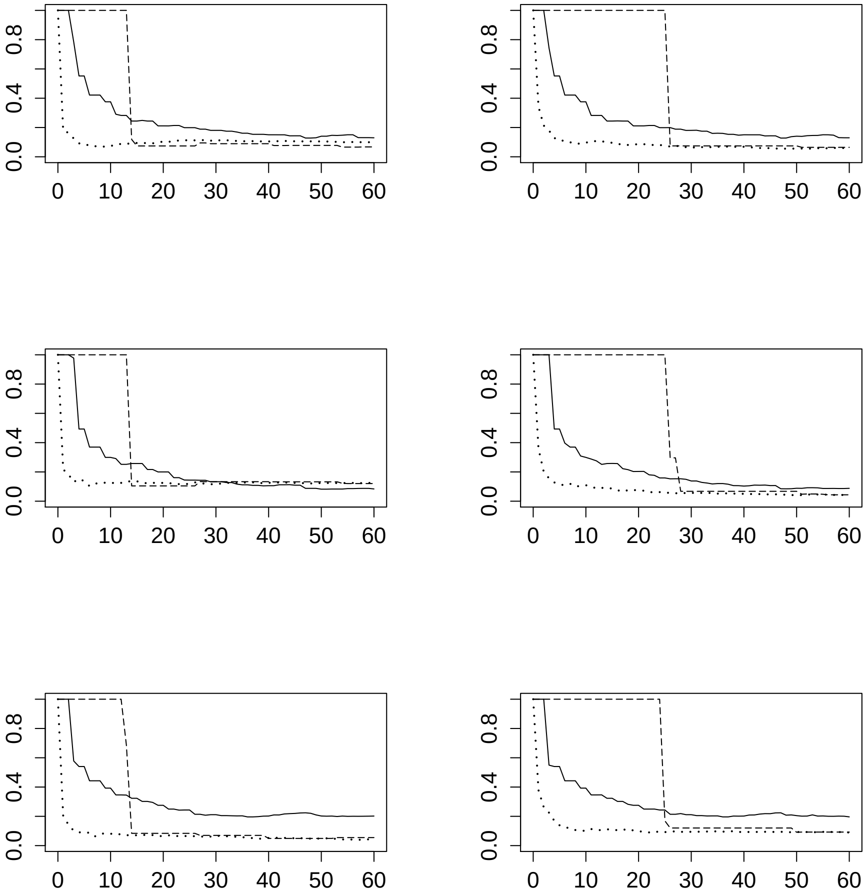


Figure 5.4: Time series regression simulation results with $\xi = 95\%$ quantile of $T_n = MSM/MSE$, sample size n=100000, dimension $d = 10$, block length L=50, autocorrelation $\rho =$ -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size $b = 5000$ (left column) ,10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 60 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

small time budgets, SDB stabilizes faster.

5.9 Discussion

In this chapter we present a new resampling method, called subsampled double bootstrap (SDB), for estimating the precision of inference methods in massive

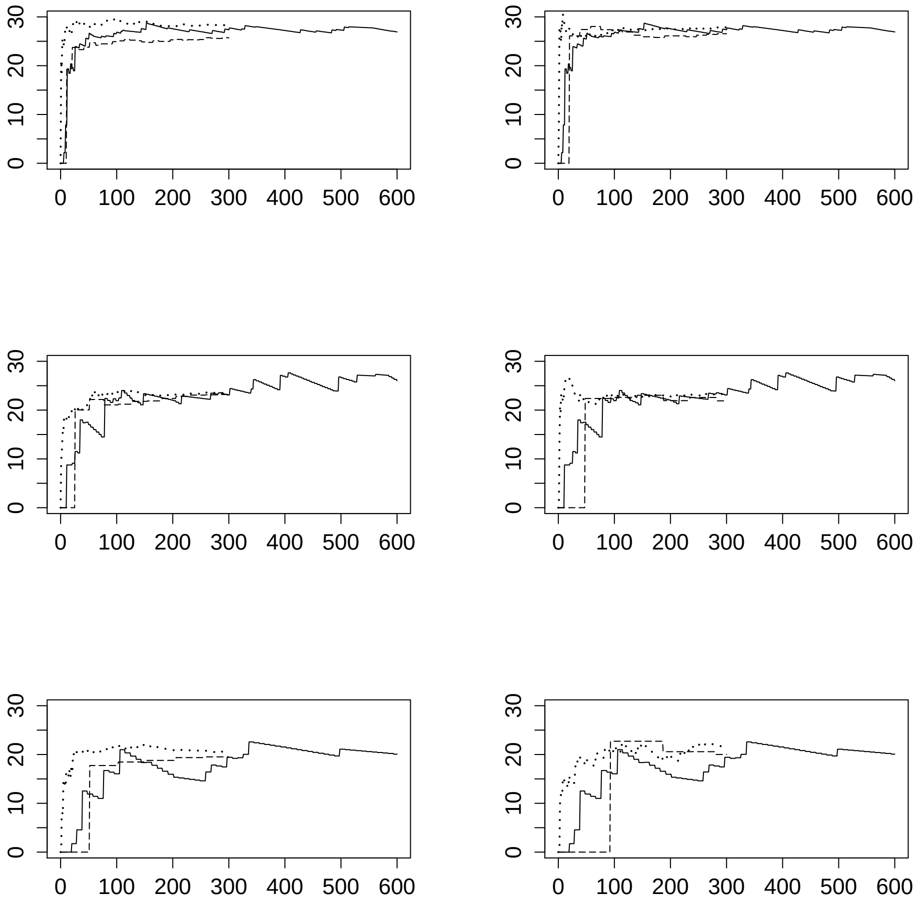


Figure 5.5: Time evolution of $\hat{\xi}$ for CET dataset (measured in Celsius), where $\xi = (q_{0.95} - q_{0.05})$ is the width of the 90% confidence interval for μ based on $T_n = \sqrt{n}(\bar{X} - \mu)$. MBB was allowed to run for 600 seconds and BLB, SDB for 300 seconds. Block length L=50 (top row), 20 (middle row), 10 (bottom row), and subset size $b = 5000$ (left column) ,10000 (right column). MBB estimates are in solid lines, BLB in dashed lines, and SDB in dottted lines.

data. Our method applies to both independent data and stationary time series data. The main idea is to select small random subsets of the data and construct a single full size resample from each random subset, in a manner reminiscent of fast double bootstrap (White (2000) and Davidson and MacKinnon (2000)). Our method inherits the theoretical strengths and automatic nature of classical resample based methods like bootstrap (Efron (1979)) in the independent data context and MBB (Künsch (1989), Liu and Singh (1992)) in the time series context. It also inherits the computational strengths of subsample based methods like subsampling (Politis and Romano (1994a)) and m out of n bootstrap (Bickel et al. (1997)). The advantage of our method over the recently proposed BLB (Kleiner et al. (2014)) lies in sample coverage, running time, and automatic implementation without having to choose additional tuning parameters under a given time budget. Simulation studies and data analysis examples demonstrate the advantage of our method over BLB and boostrap (i.i.d. case) or MBB (time series case) for a given computational time budget.

An important practical aspect of both SDB and BLB is the choice of subset size. Increasing the subset size leads to increasing benefits in terms of statistical accuracy but at an increasing computational cost. In the time series case, the regularity conditions of Theorem 5.6.1 impose some restrictions on b for consistency. In practice, for a given computational time budget, it remains unclear how to choose an optimal subset size that balances statistical accuracy and running time. A closely related problem is the selection of optimal block length for the time series version of SDB and BLB. In the context of classical resampling methods this problem has been well studied by Hall et al. (1995), Bühlmann and Künsch (1999) among others. For the time series version of SDB and BLB, we conjecture that the choice of optimal block length is associated with subset size in addition to sample size and other parameters. We leave these interesting directions to future work.

Further, it is worth mentioning that the higher order accuracy of BLB was studied by Kleiner et al. (2014) and higher order accuracy of FDB has been recently studied by Chang and Hall (2014). A relevant next step is a theoretical comparison of SDB and BLB which will help identify scenarios where SDB works better than BLB, or vice versa. This comparison will involve studying higher order properties of SDB, and we plan to consider this in future research as well.

5.10 Proofs of theoretical results

5.10.1 Proof of Theorem 5.3.1

We begin by setting up a probabilistic model for the SDB in the i.i.d. setting. When dealing with empirical processes which are defined on classes of functions, measurability questions play an important role and it is crucial to state what the underlying probability space is– see Dudley (1999), Chapter 3.1 (page 91), for a discussion of related matters. Here we will consider the following setup.

(P) Consider a product of three probability spaces $(\Omega_i^n, \mathcal{A}_i^n, P_i^n)_{i=1,...,3}$. Assume that the observations $X_1, ..., X_n$ are defined as coordinate projections on $(\Omega_1^n, \mathcal{A}_1^n, P_1^n)$, which is itself a product of n identical probability spaces [this is a standard assumption in empirical process theory- see for instance Dudley (1999), Chapter 3.1]. Additionally, assume that on Ω_2^n we have a random vector $(W_{1,n}, ..., W_{b,n}) \sim \text{Multinomial}_b(n, 1/b, ..., 1/b)$ and that on Ω_3^n we have a random variable R which follows a uniform distribution on the permutations of $\{1, ..., n\}$. In what follows, denote the set of permutations of $\{1, ..., n\}$ by $\sigma(n)$. Also, we assume without loss of generality that Ω_i^n are finite for $i = 2, 3$ and that for $i = 2, 3$ the sigma-algebra $\mathcal{A}_i^n$ is the power set of Ω_i^n.

Throughout this proof, write W for the vector $(W_{1,n}, ..., W_{b,n})$ and X for the vector $(X_1, ..., X_n)$. Define the map

$$f \mapsto (Z_n(R, X, W))(f) := \frac{1}{\sqrt{n}} \sum_{i=1}^{b} (W_i - n/b) f(X_{R^{-1}(i)}), \quad f \in \mathcal{F}.$$

Note that $(Z_n(R, X, W))(\cdot)$ can be viewed as an element of the space of functions $\ell^\infty(\mathcal{F})$. Throughout the proof, denote by BL_1 the set of Lipschitz continuous functions $g : \ell^\infty(\mathcal{F}) \to \mathbb{R}$ with Lipschitz constant 1 that are additionally uniformly bounded by 1. Also, use the notation f^*, f_* to denote smallest measurable majorants and greatest measurable minorants, respectively. For maps of several arguments, it will sometimes be necessary to take measurable majorants and minorants with respect to only some of the arguments. For example $g(r, X, W)^{*X,W}$ will be used to denote the smallest measurable majorant of the map $(x, w) \mapsto g(r, x, w)$ with r being held fixed. With this notation, we need to show that [see Kosorok (2008), Section 2.2.3]

(i)
$$\sup_{h \in \text{BL}_1} \left| \mathbb{E}_{R,W} h(Z_n(R, X, W)) - \mathbb{E}[h(\mathbb{G})] \right| \xrightarrow{\mathbb{P}^*} 0.$$

(ii) For all $h \in \text{BL}_1$

$$\mathbb{E}_{R,W} h(Z_n(R, X, W))^* - \mathbb{E}_{R,W} h(Z_n(R, X, W))_* \xrightarrow{\mathbb{P}^*} 0.$$

Here, $\mathbb{E}_{R,W}$ denotes the expectation with respect to R, W. Note that the map

$(R,W) \mapsto h(Z_n(R,X,W))$ is measurable outer almost surely since R,W are defined on complete, discrete probability spaces.

Proof of (i) Write

$$\mathbb{E}_{R,W} h(Z_n(R,X,W)) = \frac{1}{n!} \sum_{r \in \sigma(n)} \mathbb{E}_W \Big[h(Z_n(r,X,W)) \Big]$$

Then

$$\begin{aligned}
& \mathbb{E}_X^* \Big[\sup_{h \in \mathrm{BL}_1} \Big| \mathbb{E}_{R,W} h(Z_n(R,X,W)) - \mathbb{E}[h(\mathbb{G})] \Big| \Big] \\
\leq \; & \mathbb{E}_X^* \Big[\frac{1}{n!} \sum_{r \in \sigma(n)} \sup_{h \in \mathrm{BL}_1} \mathbb{E}_W \Big| h(Z_n(r,X,W)) - \mathbb{E}[h(\mathbb{G})] \Big| \Big] \\
\leq \; & \mathbb{E}_X \Big[\frac{1}{n!} \sum_{r \in \sigma(n)} \mathbb{E}_W \Big[\Big(\sup_{h \in \mathrm{BL}_1} \Big| h(Z_n(r,X,W)) - \mathbb{E}[h(\mathbb{G})] \Big| \Big)^{*X,W} \Big] \Big] \\
= \; & \frac{1}{n!} \sum_{r \in \sigma(n)} \mathbb{E}^* \Big[\sup_{h \in \mathrm{BL}_1} \Big| h(Z_n(r,X,W)) - \mathbb{E}[h(\mathbb{G})] \Big| \Big].
\end{aligned}$$

For each fixed value of r we have

$$(Z_n(r,X,W))(f) = \frac{1}{\sqrt{n}} \sum_{i=1}^{b} (W_{i,n} - n/b) f(X_{r^{-1}(i)}),$$

which implies that $Z_n(r,X,W)$ depends on X only through $(X_{r^{-1}(i)})_{i=1,\ldots,b}$. In particular

$$\sup_{h \in \mathrm{BL}_1} \Big| h(Z_n(r,x,w)) - \mathbb{E}[h(\mathbb{G})] \Big| = S \circ \Pi_r(x,w)$$

where $\Pi_r(x,w) := ((x_{r^{-1}(1)}, \ldots, x_{r^{-1}(b)}), w)$ and we defined for $y, w \in \mathbb{R}^b$

$$S(y,w) := \sup_{h \in \mathrm{BL}_1} \Big| h\Big(f \mapsto \frac{1}{\sqrt{n}} \sum_{i=1}^{b} (w_i - n/b) f(y_i) \Big) - \mathbb{E}[h(\mathbb{G})] \Big|.$$

Since $X_1, \ldots, X_n, W$ are defined on a product probability space, it follows that the measurable majorant of $S \circ \Pi_r(X,W)$ with respect to X, W can be expressed as $S(\cdot,\cdot)^* \circ \Pi_r(x,w)$. This is a consequence of from Lemma 1.2.5 in van der Vaart and Wellner (1996) and combined with the fact that S is uniformly bounded and Π_r is a coordinate projection on a product space. In particular, the symmetry of the problem implies that

$$\frac{1}{n!} \sum_{r \in \sigma(n)} \mathbb{E}^* \Big[\sup_{h \in \mathrm{BL}_1} \Big| h(Z_n(r,X,W)) - \mathbb{E}[h(\mathbb{G})] \Big| \Big] = \mathbb{E}^* \Big[\sup_{h \in \mathrm{BL}_1} \Big| h(Z_n(\mathrm{id},X,W)) - \mathbb{E}[h(\mathbb{G})] \Big| \Big]$$

where id := $(1, 2, ..., n)$. Moreover Theorem 3.6.3 in van der Vaart and Wellner (1996) with the identification $k_n = n, n = b$ implies that

$$2 \geq \sup_{h \in \mathrm{BL}_1} \Big| h(Z_n(\mathrm{id}, X, W)) - \mathbb{E}[h(\mathbb{G})] \Big| \overset{\mathbb{P}^*}{\to} 0.$$

By dominated convergence, this yields

$$\mathbb{E}^* \Big[\sup_{h \in \mathrm{BL}_1} \Big| h(Z_n(\mathrm{id}, X, W)) - \mathbb{E}[h(\mathbb{G})] \Big| \Big] \to 0,$$

and thus statement (i) is established.

Proof of (ii)
It suffices to prove that [note that $h^* - h_* \geq 0$]

$$\mathbb{E}[h(Z_n(R, X, W))^* - h(Z_n(R, X, W))_*] \to 0.$$

Observe that by the definition of measurable majorants we have

$$\begin{aligned} h(Z_n(R, X, W))^* &= \Big(\sum_{r \in \sigma(n)} I\{R = r\} h(Z_n(r, X, W)) \Big)^* \\ &\qquad \leq \sum_{r \in \sigma(n)} I\{R = r\} \Big(h(Z_n(r, X, W)) \Big)^{*X,W}, \end{aligned}$$

since $(R, X, W) \mapsto I\{R = r\}\Big(h(Z_n(r, X, W))\Big)^{*X,W}$ is measurable for each $r \in \sigma(n)$. Similarly

$$\begin{aligned} h(Z_n(R, X, W))_* &= \Big(\sum_{r \in \sigma(n)} I\{R = r\} h(Z_n(r, X, W)) \Big)_* \\ &\qquad \geq \sum_{r \in \sigma(n)} I\{R = r\} \Big(h(Z_n(r, X, W)) \Big)_{*X,W}. \end{aligned}$$

Thus

$$\begin{aligned} & \mathbb{E}[h(Z_n(R, X, W))^* - h(Z_n(R, X, W))_*] \\ \leq\ & \mathbb{E}\Big[\sum_{r \in \sigma(n)} I\{R = r\} \Big(\Big(h(Z_n(r, X, W)) \Big)^{*X,W} - \Big(h(Z_n(r, X, W)) \Big)_{*X,W} \Big) \Big] \\ =\ & \mathbb{E}\Big[\Big(h(Z_n(\mathrm{id}, X, W)) \Big)^{*X,W} - \Big(h(Z_n(\mathrm{id}, X, W)) \Big)_{*X,W} \Big] \end{aligned}$$

where the equality in the last line follows by arguments similar to the ones given in the proof of (i). Now the expression in the last line converges to zero by arguments similar to the ones given in the proof of (i) and Theorem 3.6.3 in van der Vaart and Wellner (1996) and thus (ii) follows. □

5.10.2 Proof of Theorem 5.6.1

Throughout this proof, we will simplify notation by assuming that $KL = n$. It is easy to see that this assumption can be relaxed.

Introduce the abbreviation $S = (s_1, ..., s_K), \mathcal{X} = (X_1, ..., X_n)$. Denote by $P_{S,J}$ the probability conditional on $\mathcal{X}$ and by P_S the probability conditional on $\mathcal{X}, J$. Similarly, let $\mathbb{E}_{S,J}, \mathbb{E}_S, \mathrm{Var}_{S,J}$ and Var_S denote the corresponding versions of conditional expectations and variances. Following the discussion on page 277 in Radulović (1996), we will assume that all suprema we encounter are measurable. This might not always be true, but permissibility of $\mathcal{F}$ ensures that suitable modifications of our arguments remain correct [see the discussion on page 277 in Radulović (1996)].

Define the norm $\|f\|_{p,X} := (\mathbb{E}[|f(X_1)|^p])^{1/p}$. For an arbitrary δ-net $\mathcal{F}_\delta$ for $\mathcal{F}$ with respect to $\|\cdot\|_{p,X}$, denote by f_δ any point in $\mathcal{F}_\delta$ which minimizes $\|f - g\|_{p,X}$ over $g \in \mathcal{F}_\delta$. Consider the approximating processes $\mathbb{A}^B_{\delta,n}(f) := \hat{\mathbb{G}}^B_{n,b}(f_\delta), \mathbb{A}_\delta(f) := \mathbb{G}(f_\delta)$. By Lemma B.3 in Volgushev and Shao (2014) the claim of the Theorem follows once we establish that

(i) For every $i \in \mathbb{N}$: $\mathbb{A}^B_{1/i,n} \underset{J,S}{\overset{\mathbb{P}}{\rightsquigarrow}} \mathbb{A}_{1/i}$ for $n \to \infty$.

(ii) $\mathbb{A}_{1/i} \rightsquigarrow \mathbb{G}$ for $i \to \infty$.

(iii) For every $\varepsilon > 0$: $\lim_{i\to\infty} \limsup_{n\to\infty} P(\sup_{f\in\mathcal{F}} |\mathbb{A}^B_{1/i,n}(f) - \hat{\mathbb{G}}^B_{n,b}(f)| > \varepsilon) = 0$.

Part (ii) follows from the properties of the limiting process $\mathbb{G}$ [more precisely, there exists a version of $\mathbb{G}$ with sample paths that are uniformly continuous with respect to $\|\cdot\|_{2,X}$ and thus $\|\cdot\|_{p,X}$ for $p \geq 2$ - see Theorem 2.1 in Arcones and Yu (1994)]. It thus remains to establish (i) and (iii).

In order to establish (i), it suffices to show that for any fixed, finite collection of functions $f_1, ..., f_k \in \mathcal{F}$ we have

$$(\hat{\mathbb{G}}^B_{n,b}(f_1), ..., \hat{\mathbb{G}}^B_{n,b}(f_k)) \underset{J,S}{\overset{\mathbb{P}}{\rightsquigarrow}} (\mathbb{G}(f_1), ..., \mathbb{G}(f_k)). \tag{5.1}$$

Denote by BL_1 the set of functions on $\mathbb{R}^k$ which are bounded by 1 and are Lipschitz continuous with Lipschitz constant bounded by 1. In order to establish (5.1), we need to prove that

$$\sup_{h\in \mathrm{BL}_1} \mathbb{E}_{S,J}\Big|h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| = o_P(1) \tag{5.2}$$

Due to the independence between J and $\mathcal{X}$ and due to strict stationarity of $\{X_t\}_{t\in\mathbb{Z}}$, the distribution of the tuple $(X_J,...,X_{J+b-1})$ is the same as the distribution of $(X_1,...,X_b)$ [unconditionally]. Thus the arguments on page 272 in Radulović (1996) yield

$$(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) \underset{S}{\overset{\mathbb{P}}{\rightsquigarrow}} (\mathbb{G}(f_1),...,\mathbb{G}(f_k)). \tag{5.3}$$

Observe that

$$\begin{aligned}
&\sup_{h\in \mathrm{BL}_1} \Big|\mathbb{E}_{S,J}[h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k))] - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| \\
&\qquad \le \sup_{h\in \mathrm{BL}_1} \mathbb{E}_J\mathbb{E}_S\Big|h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| \\
&\qquad\qquad \le \mathbb{E}_J \sup_{h\in \mathrm{BL}_1} \mathbb{E}_S\Big|h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big|.
\end{aligned}$$

Thus for any $\varepsilon > 0$

$$\begin{aligned}
&\mathbb{E}\sup_{h\in \mathrm{BL}_1} \Big|\mathbb{E}_{S,J}[h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k))] - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| \\
&\qquad \le \mathbb{E}\sup_{h\in \mathrm{BL}_1} \mathbb{E}_S\Big|h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| \\
&\quad \le \varepsilon + 2P\Big(\sup_{h\in \mathrm{BL}_1} \mathbb{E}_S\Big|h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| > \varepsilon\Big)
\end{aligned}$$

since $\Big|h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| \le 2$ by the definition of BL_1. By (5.3), the probability in the last line of the above equation tends to zero [as $n \to \infty$] for any fixed $\varepsilon > 0$, and this implies

$$\mathbb{E}\sup_{h\in \mathrm{BL}_1} \mathbb{E}_{S,J}\Big|h(\hat{\mathbb{G}}^B_{n,b}(f_1),...,\hat{\mathbb{G}}^B_{n,b}(f_k)) - \mathbb{E}[h(\mathbb{G}(f_1),...,\mathbb{G}(f_k))]\Big| = o(1).$$

This proves (5.2) and establishes (i).

Next, let us prove (iii). Fix $\varepsilon > 0$. For a function f, define its truncated version

$f^t(x) := f(x)I\{F(x) \le b^\gamma\}$. We shall prove that

$$\sup_{f\in\mathcal{F}} |\hat{\mathbb{G}}_{n,b}^B(f) - \hat{\mathbb{G}}_{n,b}^B(f^t)| = o_P(1). \tag{5.4}$$

Additionally, we will apply restricted chaining to show that there exists a sequence of sets of functions $\mathcal{F}_n \subset \mathcal{F}, n \in \mathbb{N}$ such that

$$\begin{aligned} &P\Big(\sup_{f,g\in\mathcal{F},\|f-g\|_{p,X}<\delta} \Big|\hat{\mathbb{G}}_{n,b}^B(f^t) - \hat{\mathbb{G}}_{n,b}^B(g^t)\Big| > 3\varepsilon\Big) \\ &\qquad \le P\Big(\sup_{f\in\mathcal{F}_n g\in\mathcal{F},\|f-g\|_{p,X}\le(\ln b)^{-3/2}} |\hat{\mathbb{G}}_{n,b}^B(f^t) - \hat{\mathbb{G}}_{n,b}^B(g^t)| \ge \varepsilon\Big) + \xi(\delta, n) \end{aligned} \tag{5.5}$$

where $\lim_{\delta\to 0}\lim_{n\to\infty} \xi(\delta, n) = 0$ and that

$$P\Big(\sup_{f\in\mathcal{F}_n,g\in\mathcal{F},\|f-g\|_{p,X}\le(\ln b)^{-3/2}} |\hat{\mathbb{G}}_{n,b}^B(f^t) - \hat{\mathbb{G}}_{n,b}^B(g^t)| \ge \varepsilon\Big) \to 0 \quad \text{as } n \to \infty. \tag{5.6}$$

Taken together, (5.4)-(5.6) imply (iii).

Before proceeding with the proof, we remark that

$$\hat{\mathbb{G}}_{n,b}^B(f) = \sum_{k=1}^K \frac{1}{\sqrt{n}} \sum_{i=1}^L \Big(f(X^*_{(k-1)L+i}) - \frac{1}{b}\sum_{j=1}^b f(X_{J+j})\Big) =: \sum_{k=1}^K V_k(f).$$

Note that by construction, the quantities $V_1(f), ..., V_K(f)$ are independent conditionally on $J, \mathcal{X}$. Moreover, for any function f which is uniformly bounded, we have that $|V_k(f)| \le 2n^{-1/2}L\|f\|_\infty$. Thus by Bernstein's inequality [Lemma 2.2.9 in van der Vaart and Wellner (1996)]

$$P_S\Big(\Big|\sum_{k=1}^K V_k(f)\Big| \ge \eta\Big) \le 2\exp\Big(-\frac{1}{2}\frac{\eta^2}{Kv^2 + 2n^{-1/2}L\|f\|_\infty \eta/3}\Big) \tag{5.7}$$

for any $v^2 \ge Var_S(V_1)$ [note that by construction $Var_S(V_1) = Var_S(V_k)$ for all k almost surely]. Now from the definition of the bootstrap and the fact that $KL = n$

$$KVar_S(V_k) = \frac{1}{b}\sum_{i=1}^b \Big(\frac{1}{L^{1/2}}\sum_{j=1}^L f(\tilde{X}_{J+i+j})\Big)^2 - \Big(\frac{1}{b}\sum_{i=1}^b \frac{1}{L^{1/2}}\sum_{j=1}^L f(\tilde{X}_{J+i+j})\Big)^2.$$

Additionally, due to the independence between J and the original sample and due to strict stationarity, the distribution of the tuple $(X_{J+1}, ..., X_{J+b})$ is the same as the distribution of $(X_1, ..., X_b)$ [unconditionally]. A close inspection of the proof of Lemma 3 in Radulović (1996) [after identifying (n, b) in the latter paper with (b, L) in our notation] shows that under the assumption $L = o(b^\rho)$ for some $0 < \rho < \frac{p-2}{2(p-1)}$ the following two claims are true:

$$A_n(\mathcal{G}_n) := (\ln b)^3 \sup_{h \in \mathcal{G}_n} \Big| \mathrm{Var}_S\Big(\frac{1}{L^{1/2}} \sum_{i=1}^{L} h^t(X_i^*)\Big) - \mathrm{Var}\Big(\frac{1}{L^{1/2}} \sum_{i=1}^{L} h^t(X_i)\Big)\Big| = o_P(1) \tag{5.8}$$

for any sequence of sets $\mathcal{G}_n \subset \mathcal{F}$ with cardinality $O(n^c)$ for some fixed $c < \infty$, and

$$B_n := (\ln b)^2 \sup_{h \in \mathcal{F}' : \|h\|_{p,X} \le (\ln b)^{-3/2}} \mathrm{Var}_S\Big(\frac{1}{L^{1/2}} \sum_{i=1}^{L} h^t(X_i^*)\Big) = o_P(1) \tag{5.9}$$

where $\mathcal{F}' := \{f - g : f, g \in \mathcal{F}\}$. Note that, when generating the subsamples, Radulović (1996) uses 'wrapping' while we do not. Following the discussion in Radulović (1996), it is easy to see that asymptotically this does not matter.

Additionally, equation (14) in Radulović (1996) implies that

$$\mathrm{Var}\Big(\frac{1}{L^{1/2}} \sum_{i=1}^{L} h^t(X_i)\Big) \le C_0 \|h^t\|_{p,X}^2 \tag{5.10}$$

for a constant C_0 which depends only on p and the mixing coefficients β.

Proof of (5.4)
Observe that

$$\sup_{f \in \mathcal{F}} |\hat{\mathbb{G}}_{n,b}^B(f) - \hat{\mathbb{G}}_{n,b}^B(f^t)| \le \frac{1}{\sqrt{n}} \Big(\sum_{i=1}^{n} F(X_i^*) I_{\{F(X_i^*) > b^\gamma\}} + \frac{1}{b} \sum_{j=1}^{b} F(X_{J+j}) I_{\{F(X_{J+j}) > b^\gamma\}} \Big).$$

By the Chebyshev inequality it suffices to show that the expectation of the right-hand side of the above inequality is $o(1)$. From the definition of the bootstrap, it is not difficult to see that for any $i = 1, ..., n$

$$0 \le \mathbb{E}[F(X_i^*) I\{F(X_i^*) > b^\gamma\}] = \mathbb{E}[F(X_1) I\{F(X_1) > b^\gamma\}] \le \|F\|_{p,X} (P(F(X_1) > b^\gamma))^{\frac{p-1}{p}}$$

and the right-hand side of the equation above is of order $o(b^{-(p-1)\gamma})$ by dominated convergence. A similar bound holds for $\mathbb{E}[F(X_{J+j}) I\{F(X_{J+j}) > b^\gamma\}]$. Thus (5.4) follows from the condition $n^{1/2} = O(b^{(p-1)\gamma})$.

Proof of (5.5)

Define $\psi_1(x) := e^x - 1$ and denote by $\|\cdot\|_{\psi_1}$ the corresponding *Orlitz norm* [see van der Vaart and Wellner (1996), Chapter 2.2]. Fix $\delta > 0$ and let k_n denote the smallest integer such that $\delta/2^{k_n} < (\ln b)^{-3/2}/2$. Successively construct sets $\mathcal{G}_1 \subset \mathcal{G}_2 \subset ... \subset \mathcal{G}_{k_n}$ which are maximal subsets of $\mathcal{F}$ with the property $\|f-g\|_{p,X} \geq 2^{-i}\delta$ for all $f, g \in \mathcal{G}_i$ [here, maximal means that no further element can be added to $\mathcal{G}_i$ without destroying the property that $\|f-g\|_{p,X} \geq 2^{-i}\delta$ for all $f, g \in \mathcal{G}_i$]. Observe that the cardinality of $\mathcal{G}_{k_n}$ is of polynomial order in $2^{-k_n}\delta$ [the cardinality of $\mathcal{G}_{k_n}$ is bounded by the packing number, which is polynomial since $\mathcal{F}$ is VC- see Theorem 2.6.7 and the discussion on page 98 in van der Vaart and Wellner (1996)], and thus of polynomial order in n for any fixed δ. Set $\alpha(n) := 2^{-k_n}\delta$ and identify the set $\mathcal{F}_n$ with $\mathcal{G}_{k_n}$. Next, define the event $D_n := \{A_n(\mathcal{F}_n) \leq 1\}$ and note that $P(D_n) \to 1$ for $n \to \infty$ [recall the definition of A_n in (5.8)] for any fixed δ, this follows from (5.8). Observe that I_{D_n} is independent of S and that by definition of D_n we have for any $f \in \mathcal{F}_{\alpha(n)}$ and any $\eta > 0$

$$\begin{aligned}
P\Big(|\hat{\mathbb{G}}^B_{n,b}(f^t)|I_{D_n} > \eta\Big) &= \mathbb{E}\mathbb{E}_S I_{\{|\hat{\mathbb{G}}^B_{n,b}(f^t)|>\eta\}} I_{D_n} \\
&\leq 2\mathbb{E}\Big[I_{D_n}\exp\Big(-\frac{1}{2}\frac{\eta^2}{\mathrm{Var}_S\Big(L^{-1/2}\sum_{i=1}^L h(X_i^*)\Big) + \frac{2}{3}n^{-1/2}Lb^\gamma\eta}\Big)\Big] \\
&\leq 2\mathbb{E}\Big[I_{D_n}\exp\Big(-\frac{1}{2}\frac{\eta^2}{C_0\|f^t\|^2_{p,X} + (\ln b)^{-3} + \frac{2}{3}n^{-1/2}Lb^\gamma\eta}\Big)\Big] \\
&\leq 2\exp\Big(-\frac{1}{2}\frac{\eta^2}{C_0\|f^t\|^2_{p,X} + (\ln b)^{-3} + \frac{2}{3}n^{-1/2}Lb^\gamma\eta}\Big)
\end{aligned}$$

where the first inequality follows from (5.7) and the second from the definition of D_n. From the inequality above combined with Lemma 2.2.10 in van der Vaart and Wellner (1996) [applied with $m = 1$] we obtain that for any $f \in \mathcal{F}_n$

$$\Big\|\hat{\mathbb{G}}^B_{n,b}(f^t)I_{D_n}\Big\|_{\psi_1} \leq C\Big(n^{-1/2}Lb^\gamma + (\|f^t\|^2_{p,X} + (\ln b)^{-3})^{1/2}\Big)$$

for some constant C that does not depend on f, δ, n. In particular we obtain for sufficiently large n [due to the assumptions on L, b, n]

$$\Big\|\hat{\mathbb{G}}^B_{n,b}(f^t)I_{D_n}\Big\|_{\psi_1} \leq C'\|f^t\|_{p,X} \quad \forall f \in \mathcal{F}_n : \|f^t\|_{p,X} \geq (\ln b)^{-3/2}/2. \tag{5.11}$$

We now shall apply Lemma 7.1 from Kley et al. (2014). In the notation of that Lemma, let $T := \mathcal{F}, d(f,g) := \|f^t - g^t\|_{p,X}, \bar{\eta} := (\ln b)^{-3/2}, \Psi := \psi_1, \eta = \delta, \mathbb{G}_f :=$

$\hat{\mathbb{G}}_{n,b}^B(f^t)I_{D_n}$. A careful inspection of the proof of that Lemma reveals that (5.11) is already sufficient to obtain the bound

$$\sup_{f,g\in\mathcal{F}:\|f-g\|_{p,X}\leq\delta}|\hat{\mathbb{G}}_{n,b}^B(f^t)-\hat{\mathbb{G}}_{n,b}^B(g^t)|I_{D_n}$$
$$\leq S_1+2\sup_{f\in\mathcal{F}_n,g\in\mathcal{F},\|f-g\|_{p,X}\leq(\ln b)^{-3/2}}|\hat{\mathbb{G}}_{n,b}^B(f^t)-\hat{\mathbb{G}}_{n,b}^B(g^t)|I_{D_n}$$

where S_1 is such that [note that $\psi_1^{-1}(x)=\ln(1+x)$ and that the packing number of $\mathcal{F}$ with respect to $\|\cdot\|_{p,X}$ is of polynomial order since $\mathcal{F}$ is VC- see Theorem 2.6.7 and the discussion on page 98 in van der Vaart and Wellner (1996)]

$$\|S_1\|_{\psi_1}\leq C\Big[\int_{(\ln b)^{-3/2}/2}^{\delta}1+|\log\varepsilon|d\varepsilon+(\delta+2(\ln b)^{-3/2})(1+|\log\delta|)\Big]$$

for some constant C independent of δ,n. To complete the proof of (5.5), observe that

$$P\Big(\sup_{f,g\in\mathcal{F},\|f-g\|_{p,X}<\delta}\Big|\hat{\mathbb{G}}_{n,b}^B(f^t)-\hat{\mathbb{G}}_{n,b}^B(g^t)\Big|>3\varepsilon\Big)$$
$$\leq P\Big(\sup_{f,g\in\mathcal{F},\|f-g\|_{p,X}<\delta}\Big|\hat{\mathbb{G}}_{n,b}^B(f^t)-\hat{\mathbb{G}}_{n,b}^B(g^t)\Big|I_{D_n}>3\varepsilon\Big)+1-P(D_n)$$
$$\leq P(|S_1|>\varepsilon)+P\Big(\sup_{f\in\mathcal{F}_n,g\in\mathcal{F},\|f-g\|_{p,X}\leq(\ln b)^{-3/2}}|\hat{\mathbb{G}}_{n,b}^B(f^t)-\hat{\mathbb{G}}_{n,b}^B(g^t)|\geq\varepsilon\Big)$$
$$+1-P(D_n).$$

Setting $\xi(n,\delta):=P(|S_1|>\varepsilon)+1-P(D_n)$ completes the proof of (5.5).

Proof of (5.6)
Define $P_nf:=\frac{1}{n}\sum_{i=1}^n f(X_i)$ and consider the pseudo-distance $d_n(f,g):=P_n|f-g|$. Observe that to each $f\in\mathcal{F}$ we can attach a $\tilde{f}\in\mathcal{F}_n$ such that $\|f-\tilde{f}\|_{p,X}\leq(\ln b)^{-3/2}$. Let $\mathcal{H}:=\{f^t-\tilde{f}^t:f\in\mathcal{F}\}$ and denote by $\mathcal{H}_n$ an n^{-2} net for $\mathcal{H}$ under d_n. To each $h\in\mathcal{H}$, attach a $\tilde{h}\in\mathcal{H}$ such that $d_n(h,\tilde{h})\leq n^{-2}$. Since $\mathcal{F}$ is VC, $\mathcal{H}_n$ can be chosen such that, for n sufficiently large, the cardinality of $\mathcal{H}_n$ is bounded by $C(P_nF)^cn^c$ for some fixed constants C,c which do not depend on J,S. Define the event [recall the definition of B_n in (5.10)]

$$\tilde{D}_n:=\Big\{B_n\vee\Big|n^{-1}\sum_{i=1}^n F(X_i)-\mathbb{E}[F(X_1)]\Big|\leq 1\Big\}$$

and note that by (5.10) $P(\tilde{D}_n) \to 1$ [since under the assumptions of the present Theorem $P_nF - \mathbb{E}F[X_1] \to 0$ in probability] and that $I_{\tilde{D}_n}$ is independent of S. Observe that

$$\sup_{f\in\mathcal{F}_n, \|f-g\|_{p,X}\le(\ln b)^{-3/2}} |\hat{\mathbb{G}}^B_{n,b}(f^t) - \hat{\mathbb{G}}^B_{n,b}(g^t)| \le \sup_{h\in\mathcal{H}} \left|\hat{\mathbb{G}}^B_{n,b}(\tilde{h})\right| + \sup_{h\in\mathcal{H}} \left|\hat{\mathbb{G}}^B_{n,b}(h-\tilde{h})\right|$$

and that for any function f

$$\begin{aligned}\left|\hat{\mathbb{G}}^B_{n,b}(f)\right| &\le \frac{1}{\sqrt{n}}\sum_{i=1}^{n}|f(X_i^*)| + \frac{\sqrt{n}}{b}\sum_{i=0}^{b-1}|f(X_{J+i})| \le \frac{n^2}{\sqrt{n}}\frac{1}{n}\sum_{i=1}^{n}|f(X_i)| + \frac{n\sqrt{n}}{b}\frac{1}{n}\sum_{i=1}^{n}|f(X_i)|\\ &\le 2n^{3/2}\frac{1}{n}\sum_{i=1}^{n}|f(X_i)|.\end{aligned}$$

Thus by definition of $\tilde{h}$

$$\sup_{h\in\mathcal{H}}\left|\hat{\mathbb{G}}^B_{n,b}(h-\tilde{h})\right| \le 2n^{3/2}\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}|(h-\tilde{h})(X_i)| \le 2n^{-1/2}.$$

Hence it suffices to show $\sup_{h\in\mathcal{H}}\left|\hat{\mathbb{G}}^B_{n,b}(\tilde{h})\right| = o_P(1)$. To this end, note that on $\tilde{D}_n$ the cardinality of $\mathcal{H}_n$ is bounded by $C'n^{c'}$ for some constants C', c' independent of

n. Thus

$$
\begin{aligned}
&P\Big(\sup_{h\in\mathcal{H}}\Big|\hat{\mathbb{G}}^B_{n,b}(\tilde{h})\Big| > \tau\Big)\\
&\le P\Big(\Big\{\sup_{h\in\mathcal{H}_n}\Big|\hat{\mathbb{G}}^B_{n,b}(h)\Big| > \tau\Big\}\cap \tilde{D}_n\Big) + 1 - P(\tilde{D}_n)\\
&= \mathbb{E}\mathbb{E}_S\Big[I_{\{\sup_{h\in\mathcal{H}_n}|\hat{\mathbb{G}}^B_{n,b}(h)|>\tau\}}I_{\tilde{D}_n}\Big] + o(1)\\
&\le \mathbb{E}\mathbb{E}_S\Big[\sum_{h\in\mathcal{H}_n} I_{\{|\hat{\mathbb{G}}^B_{n,b}(h)|>\tau\}}I_{\tilde{D}_n}\Big] + o(1)\\
&= \mathbb{E}\Big[I_{\tilde{D}_n}\sum_{h\in\mathcal{H}_n}\mathbb{E}_S[I_{\{|\hat{\mathbb{G}}^B_{n,b}(h)|>\tau\}}]\Big] + o(1)\\
&= C'n^{c'}\mathbb{E}\Big[\sup_{h\in\mathcal{H}} I_{\tilde{D}_n}P_S(|\hat{\mathbb{G}}^B_{n,b}(h)| > \tau)\Big] + o(1)
\end{aligned}
$$

by (5.7)

$$
\begin{aligned}
&\le 2C'n^{c'}\mathbb{E}\Big[\sup_{h\in\mathcal{H}}\exp\Big(-\frac{1}{2}\frac{\tau^2}{\mathrm{Var}_S\Big(L^{-1/2}\sum_{i=1}^L h(X_i^*)\Big)+\frac{4}{3}n^{-1/2}Lb^\gamma\tau}\Big)I_{\tilde{D}_n}\Big] + o(1)\\
&\le 2C'n^{c'}\mathbb{E}\Big[\exp\Big(-\frac{1}{2}\frac{\tau^2}{\sup_{h\in\mathcal{H}}\mathrm{Var}_S\Big(L^{-1/2}\sum_{i=1}^L h(X_i^*)\Big)+\frac{4}{3}n^{-1/2}Lb^\gamma\tau}\Big)I_{\tilde{D}_n}\Big] + o(1)
\end{aligned}
$$

by the definition of $\mathcal{H}$ and $\tilde{D}_n$

$$
\begin{aligned}
&\le 2C'n^{c'}\mathbb{E}\Big[\exp\Big(-\frac{1}{2}\frac{\tau^2}{(\ln b)^{-2}+\frac{4}{3}n^{-1/2}Lb^\gamma\tau}\Big)I_{\tilde{D}_n}\Big] + o(1)\\
&\le 2C'n^{c'}\exp\Big(-\frac{1}{2}\frac{\tau^2}{(\ln b)^{-2}+\frac{4}{3}n^{-1/2}Lb^\gamma\tau}\Big) + o(1)
\end{aligned}
$$

since $(\ln n)^2 = o(n^{-1/2}Lb^\gamma)$ by the assumptions on L, b, n

$= o(1)$.

This shows that $\sup_{h\in\mathcal{H}}\Big|\hat{\mathbb{G}}^B_{n,b}(\tilde{h})\Big| = o_P(1)$ and completes the proof of (5.6). □

5.11 Supplementary simulation results

In this supplementary document we present time series simulation results for $l = 10, 20$. The simulation settings are described in Section 5.7.

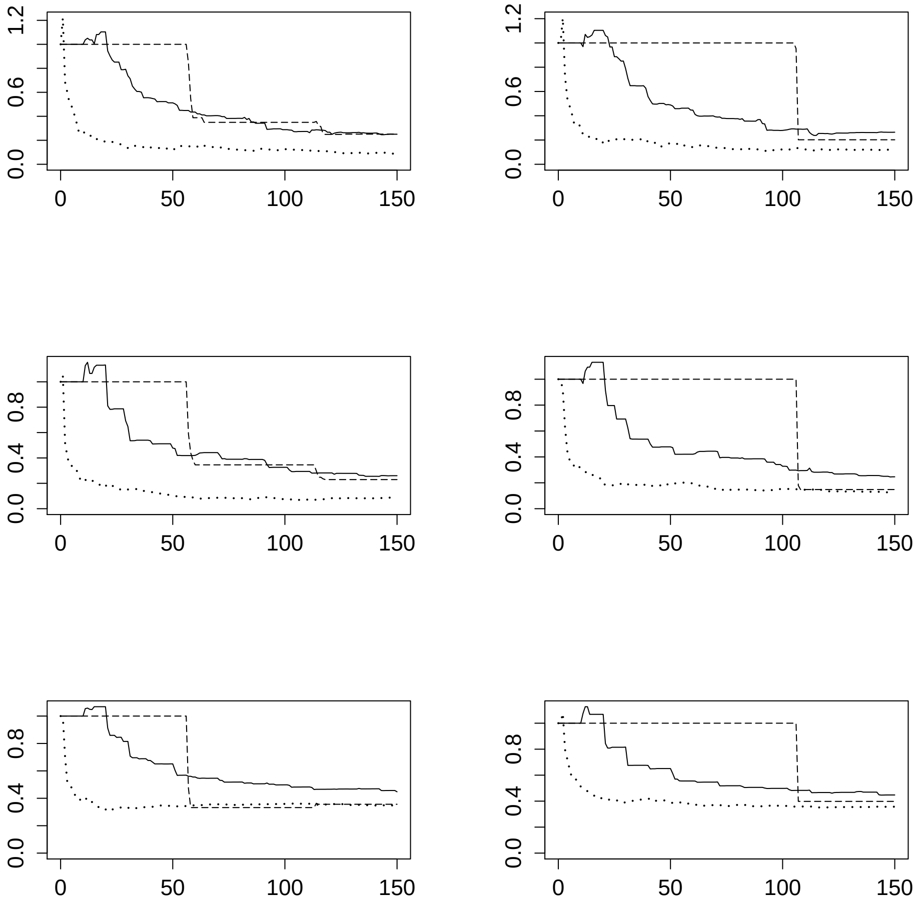


Figure 5.6: AR(1) simulation results with ξ = 95% quantile of $T_n = \sqrt{n}(M_n - M)$, sample size n=100000, block length L=10, autocorrelation ρ = -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size b = 5000 (left column) ,10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 120 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

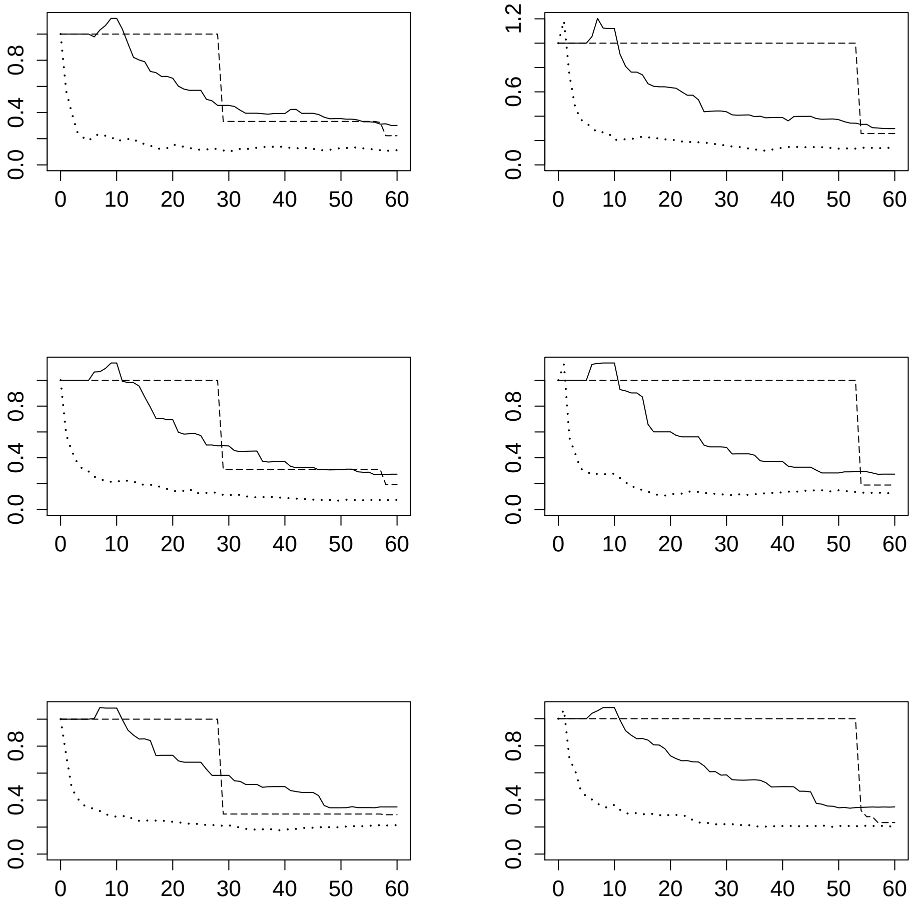


Figure 5.7: AR(1) simulation results with ξ = 95% quantile of $T_n = \sqrt{n}(M_n - M)$, sample size n=100000, block length L=20, autocorrelation ρ = -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size b = 5000 (left column) ,10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 120 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

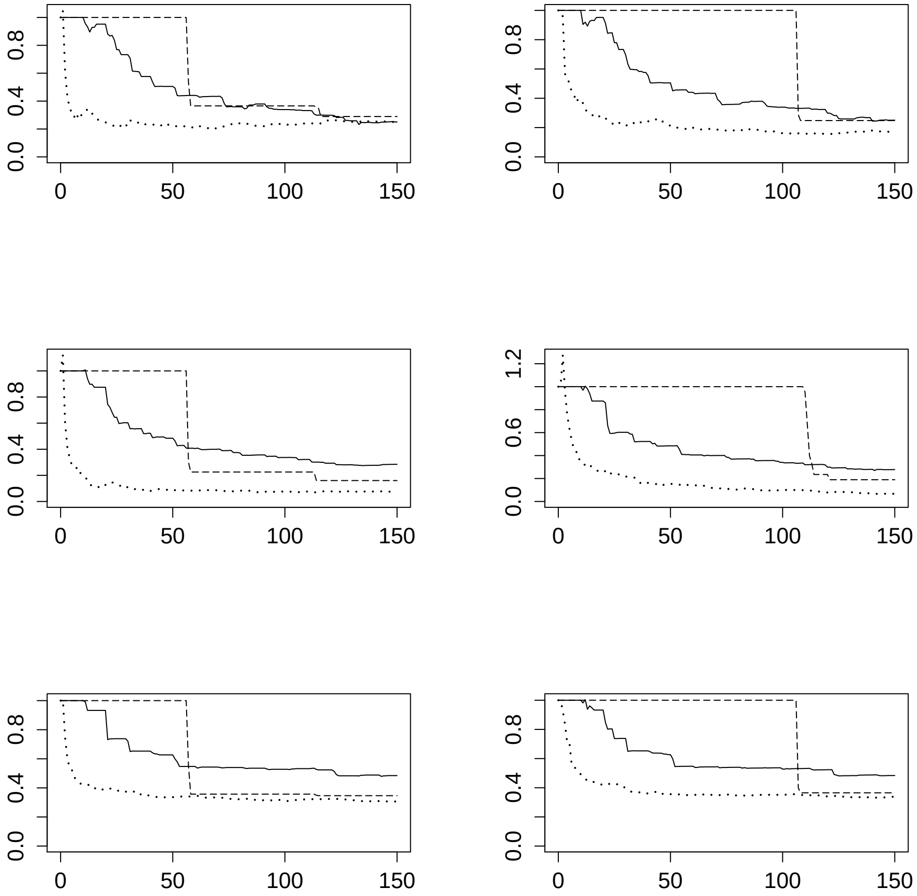


Figure 5.8: AR(1) simulation results with $\xi = 5\%$ quantile of $T_n = \sqrt{n}(M_n - M)$, sample size n=100000, block length L=10, autocorrelation ρ = -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size b = 5000 (left column) ,10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 120 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

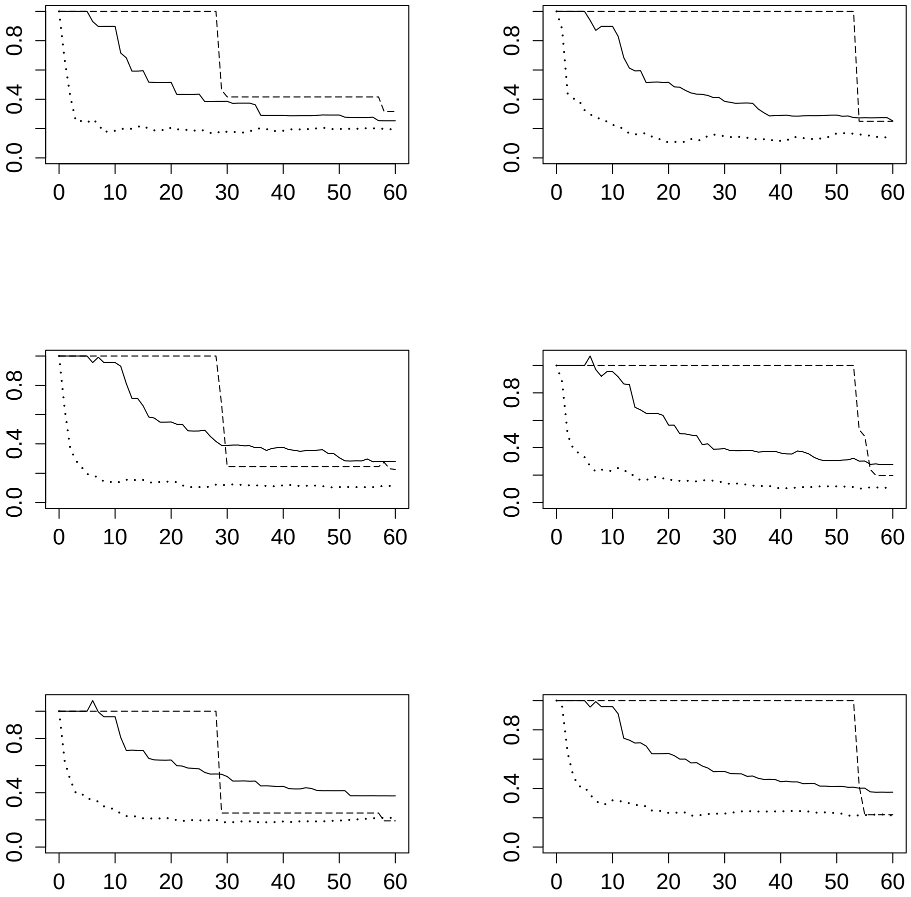


Figure 5.9: AR(1) simulation results with $\xi = 5\%$ quantile of $T_n = \sqrt{n}(M_n - M)$, sample size n=100000, block length L=20, autocorrelation ρ = -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size b = 5000 (left column) ,10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 120 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

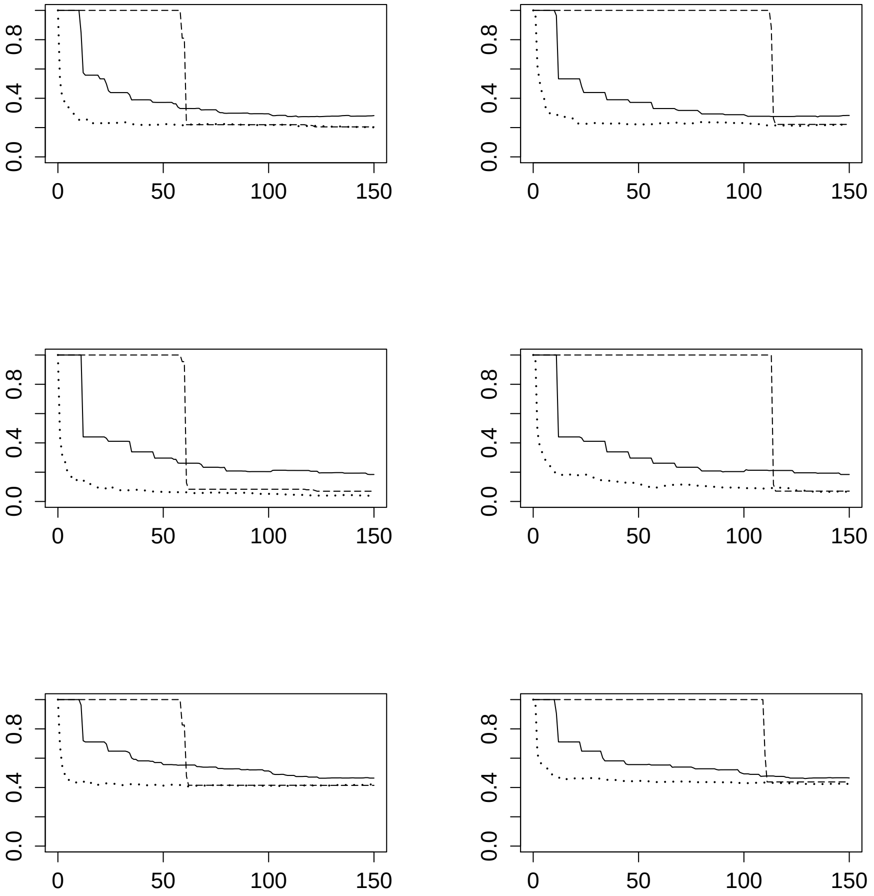


Figure 5.10: Time series regression simulation results with $\xi = 95\%$ quantile of $T_n = MSM/MSE$, sample size n=100000, dimension $d = 10$, block length L=10, autocorrelation ρ = -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size b = 5000 (left column), 10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 150 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

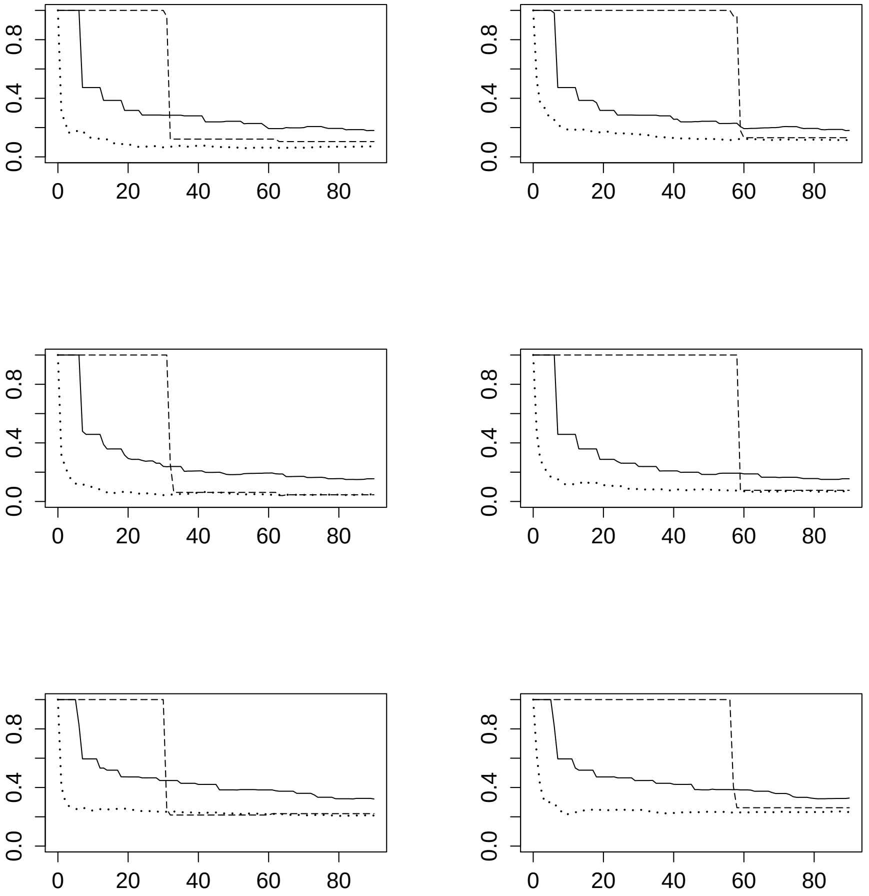


Figure 5.11: Time series regression simulation results with $\xi = 95\%$ quantile of $T_n = MSM/MSE$, sample size n=100000, dimension $d = 10$, block length L=20, autocorrelation ρ = -0.8 (top row), 0.5 (middle row), 0.9 (bottom row), and subset size b = 5000 (left column), 10000 (right column). The plot displays the time evolution of error rates from 20 simulations when each method was allowed to run for 90 seconds. MBB errors are in solid lines, BLB in dashed lines, and SDB in dottted lines.

CHAPTER 6

RESAMPLING STRATEGIES FOR BIG DATA

Big data challenges traditional statistical methodology because of its *massiveness* and its *complex structure.* The challenge gets compounded by the increasingly ambitious goals set forth by the users of big data: meteorologists want to make more detailed forecasts, internet search engines want to produce better results, online merchants want to make more effective recommendations, and they want outputs within increasingly sharp *time constraints.* Bootstrap methods are especially sensitive to big data issues, since they are computationally intensive and have wide applicability in data analysis.

In this chapter, we introduce two new resampling strategies for big data — namely aggregation of little bootstraps or ALB, and subsampled residual bootstrap or SDB. The ALB is a fast bootstrap method in the same manner as BLB or SDB, and it can be implemented on a wide variety of inferential settings. On the other hand, with the SRB we focus on the specific and important context of massive regression problems, where we utilize the structure of the OLS estimator to reduce the computational expense.

6.1 Aggregation of little bootstraps

Consider an i.i.d. sample $\mathcal{X}_n = \{X_1, \ldots, X_n\}$ drawn from some unknown distribution P. The parameter of interest is $\theta = \theta(P)$, for which an estimate $\hat{\theta}_n = \hat{\theta}(\mathcal{X}_n)$ is obtained from the sample. Having chosen the estimator, the statistician often seeks to obtain further information regarding the precision of the estimator $\hat{\theta}(\mathcal{X}_n)$. This requires the estimation of some measure involving the sampling distribution of $\hat{\theta}(\mathcal{X}_n)$ and the true value of the parameter θ. From the previous chapter, recall the root function $T_n(\hat{\theta}_n, \theta)$, and measure of precision $\xi(Q_n)$ where $Q_n = Q_n(P)$ is the unknown sampling distribution of T_n.

In this section, we will analyze the following general procedure which can be viewed as extension [and modification] of both, SDB and BLB. We call this method aggregation of little bootstraps, or ALB.

1. For $s = 1, ..., S$, sample a random sample $\mathcal{X}_b^{*,s}$ of size b from $\mathcal{X}_n$.
2. For each $s = 1, ..., S, r = 1, ..., R$ sample a random sample $\mathcal{X}_n^{**,s,r}$ from $\mathcal{X}_b^{*,s}$.
3. Compute $T_n^{**,s,r} := \sqrt{n}(\hat{\theta}(\mathcal{X}_n^{**,s,r}) - \hat{\theta}(\mathcal{X}_b^{*,s}))$.
4. Compute the empirical cdf of the pooled sample $\{T_n^{**,s,r}\}_{s=1,...,S,r=1,...,R}$ and compute the precision measure based on that cdf.

For $R = 1$, this is SDB. For R 'large' this is in the spirit of BLB. However, there is one important difference. BLB suggests to compute the precision measure on every subset $\{T_n^{**,s,r}\}_{r=1,...,R}$ and take an average in the end. Whereas in ALB, we 'aggregate' all resampled roots together, and use this aggregate ensemble to compute one single precision measure.

6.1.1 Simulation results

We now report some preliminary simulation results for fast resampling procedures. We compare four resampling techniques — conventional bootstrap, bag of little bootstraps or BLB, (Kleiner et al. (2014)), aggregation of little bootstraps or ALB, and subsampled double bootstrap or SDB.

We use the same regression model that was used in Chapter 5. Consider a d-dimensional multiple linear regression model

$$y_i = \beta_1 x_{i,1} + \ldots + \beta_d x_{i,d} + e_i$$

for $i = 1, \ldots, n$. Our parameter of interest is the d-dimensional vector of slope coefficients, whose true value is $\beta = (\beta_1, ..., \beta_d) = (1, \ldots, 1)'$. We use the usual OLS estimator $\hat{\beta}$. We also want to construct a simultaneous 95% confidence region for β. Traditionally we use the F-statistic

$$T_n(\hat{\beta}, \beta) = \frac{(\hat{\beta} - \beta)' X' X (\hat{\beta} - \beta)/d}{(y - X\hat{\beta})'(y - X\hat{\beta})/(n - d - 1)}$$

to construct the joint confidence region. Let $q_{0.95}$ be the 95% quantile of the true (unknown) distribution of $T_n(\hat{\beta}, \beta)$. Then the confidence region is given by $\{\beta : T_n(\hat{\beta}, \beta) \leq q_{0.95}\}$. In general the true distribution of T_n, and hence its quantile $q_{0.95}$, is unknown. But it can be estimated by the resampling techniques described in the previous section, with $\xi(Q_n) = q_{0.95}$ where Q_n is the true distribution of T_n.

We generate $x_{i,j} \overset{iid}{\sim} t_3$ and $e_i \overset{iid}{\sim} N(0, 100)$ independently. For normally distributed errors, we know that $T_n \sim F(d, n-d-1)$, and hence the true quantiles are given by those of the corresponding F distribution. We define the error rate as

$$|\frac{\hat{q}_{0.95}}{q_{0.95}} - 1|$$

where $\hat{q}$ and q represent the estimated and true quantiles of T_n, respectively.

We want to compare the performance of BLB, ALB, and SDB (and also bootstrap for benchmark) as the number of subsets, S, and the number of resamples per subset, R, vary. Note that for SDB, R is fixed at $R = 1$. We let n=100000, d=100 and used two subset sizes, $b = n^{\gamma}$ with $\gamma = 0.7, 0.8$. With larger subset size, one should require less number of resamples and subsets. For $b = n^{0.7}$ we let $S = 250$ for SDB, and consider the cases $(S, R) = \{(1, 499), (2, 249), \ldots, (10, 49)\}$ for BLB and SDB. For $b = n^{0.8}$ we let $S = 100$ for SDB, and consider the cases $(S, R) = \{(1, 199), (2, 99), \ldots, (5, 39)\}$ for BLB and SDB. To ensure that bootstrap runtime does not exceed runtime for other methods by a lot, we perform $R = 15, 20$ bootstrap iterations respectively.

We first look at the *final estimates* obtained for each resampling method at the end of the run. Note that this is different from time evolution of errors, where we look at estimates obtained at intermediate points. In the final error viewpoint we are interested only in the accuracy achieved at the end of the pre-determined number of subsets and resamples, and the computing cost associated with this accuracy.

These numbers are reported in Tables 1 and 2 for $b = n^{0.7}, n^{0.8}$ respectively, where statistical accuracy is measured by error rate and computing cost is measured by time taken. Errors are reported in a scale of 10^{-3} for visual convenience. First, it is clear that ALB, BLB, and SDB all perform much better than bootstrap as expected, typically achieving about half errors at smaller computing cost. Between the fast methods, SDB performs approximately as good as the average BLB. Note that although in certain settings BLB performs better than the SDB, this does not contradict the conclusions of our SDB paper, because in that paper we claimed that SDB has advantages for *small* time budgets, not that final estimates from SDB are always better than BLB. The advantages of SDB over BLB are when we look at *intermediate* estimates, and that persists in this simulation study, as we will see shortly. The BLB numbers also re-affirm the issue of its accuracy varying with S and R.

For any given setting, the ALB performs better than the BLB almost always, and this advantage is substantial for smaller values of R. Heuristically, small values of R (say $R = 40, 50, 60$) imply each subset estimate is obtained from a small ensemble, and this leads to larger errors in each subset estimate. BLB simply averages over these subset estimates and hence larger errors persist in the overall BLB estimate. In contrast, ALB aggregates the resample roots from the different subsets to obtain a single estimate, and therefore the ALB estimate is less affected by errors from individual subsets. The ALB also performs better than the SDB with respect to final estimate accuracy. The only possible concern about the ALB would be that its accuracy seems to vary with values of S and R just like BLB, which is not ideal. In particular, when the subset size changes from $b = n^{0.7}$ to $b = n^{0.8}$, the accuracy of ALB shows a reversal in trend. For $b = n^{0.7}$ more subsets are better even with less resamples per subset, whereas for $b = n^{0.8}$ the converse holds. Although with only 50 simulations, we should not read too much into these trends, and we can look at a simulation study with 100 or 500 simulations to be more diligent.

To compare across subset size b, number of subsets S and number of resamples per subset R, for SDB, BLB, and ALB, in Table 1 we have $b = n^{0.8}, S(R+1) = 200$, in Table 2 we have $b = n^{0.7}, S(R+1) = 200$, and in Table 3 we have $b = n^{0.7}, S(R+1) = 500$.

Method	bootstrap	ALB				
(S,R)	R=20	(1,199)	(2,99)	(3,66)	(4,49)	(5,39)
Error $\times 10^3$	43	17	19	17	20	21
Time	62.72	54.92				

Method	SDB	BLB				
(S,R)	S=100	(1,199)	(2,99)	(3,66)	(4,49)	(5,39)
Error $\times 10^3$	24	17	22	18	25	26
Time	54.74	54.92				

Table 6.1: Results for $b = n^{0.8}$ after full run

Figure 6.1 shows the time evolution of errors, where BLB cases can be identified as $S = 1$ being the rightmost vertical, then $S = 2$ to its left, and so on. In this *intermediate accuracy* viewpoint, SDB is better than BLB, ALB (almost hidden under SDB) is slightly better than SDB at some time points, and bootstrap is worst.

Method	bootstrap	ALB				
(S,R)	R=20	(1,199)	(2,99)	(3,66)	(4,49)	(5,39)
Error $\times 10^3$	43	23	19	19	21	18
Time	69.1	17.09				

Method	SDB	BLB				
(S,R)	S=100	(1,199)	(2,99)	(3,66)	(4,49)	(5,39)
Error $\times 10^3$	26	23	22	22	26	24
Time	17.08	17.09				

Table 6.2: Results for $b = n^{0.7}$ after full run, with same (S,R) values as in Table 2

Method	bootstrap	ALB				
(S,R)	R=15	(2,249)	(4,124)	(6,83)	(8,62)	(10,49)
Error $\times 10^3$	45	17	15	14	13	13
Time	46.21	33.26				

Method	SDB	BLB				
(S,R)	S=250	(2,249)	(4,124)	(6,83)	(8,62)	(10,49)
Error $\times 10^3$	18	18	16	17	19	21
Time	33.81	33.26				

Table 6.3: Results for $b = n^{0.7}$ after full run

6.2 Subsampled residual bootstrap

Consider a linear regression model

$$y = X\beta + \epsilon$$

where ϵ is a random error vector of length n, X is the n-by-p design vector, and β is a p-by-1 vector of coefficients. We are interested in estimating the sampling distribution of some root function $T_n(\hat{\beta}, \beta)$ where $\hat{\beta}$ is an estimator of β — for illustration let's say the OLS estimator $\hat{\beta} = (X'X)^{-1}X'y$. One way to estimate this unknown sampling distribution is by obtaining an ensemble of resample estimates $\{\beta^*\}$. For this we need a resampling strategy to obtain resamples from the observed data $(y_i, \mathbf{x}_i)_{i=1}^n$.

In paired bootstrap, we assumes that the pairs $(y_i, \mathbf{x}_i)$ are i.i.d., and hence we can resample the paired observations directly. However, it might not be reasonable to assume $\mathbf{x}_i$ to be i.i.d. in some situations, for example when X is a fixed design matrix for an experiment and not a random realization. In such a situation, the reasonable approach will be to resample from the sample *conditional* on X, i.e. keeping X unchanged in the resamples. A widely used strategy in this direction is

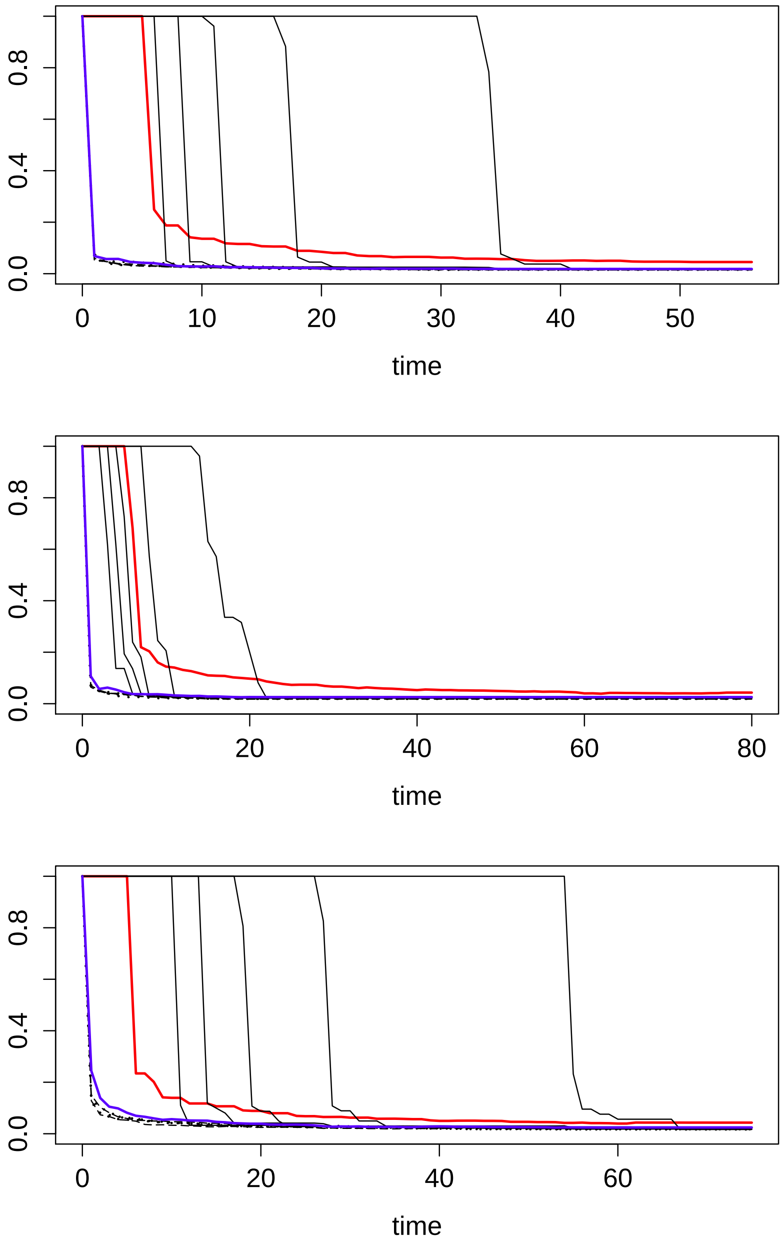


Figure 6.1: Time evolution of error rates for multiple linear regression with d=100, averaged over 50 simulations. Sample size $n = 10^5$, subset size is $b = n^{0.7}$ (top, middle), $b = n^{0.8}$ (bottom). Bootstrap errors are in solid red, SDB in solid blue, BLB in solid black, and ALB errors in dotted black. For $b = n^{0.7}$, we have $S(R+1) = 500$ in the top row, $S(R+1) = 200$ in the middle row. For clarity we have only plotted $S = 1, 2, 3, 4, 5$ for $b = n^{0.7}$ in the top row.

residual bootstrap (RB).

Let $\hat{\beta}$ be the estimated value and $\hat{y} = X\hat{\beta}$ be the fitted value of y, and $e = y - \hat{y}$ be the vector of residuals. The central idea of resampling is that i.i.d. objects can exchange places. In residual bootstrap, this implies that even when the design matrix is non-random making it unreasonable to resample y_i or $\mathbf{x}_i$, the errors are i.i.d. and hence it is reasonable to resample $\mathbf{e} = \{e_i\}_{i=1}^n$. Let $\mathbf{e}^*$ be a resample of $\mathbf{e}$, then define $y^* = \hat{y} + e^*$ as a resample of the response vector. Keeping the design matrix X unchanged, we define the resample estimate as $\beta^* = (X'X)^{-1}X'y^*$. We can repeat the resampling step R times to obtain R resamples of this form and thereby construct an ensemble of resample estimates. A potential concern is that of computational infeasibility, particularly in the context of massive data. Obtaining each resample estimate involves a calculation of order n which can be very expensive. Therefore there is a need of faster methods like subsampled double bootstrap (SDB) for RB with massive data. In its basic form SDB cannot be easily extended to RB, however we propose a modified version of SDB which might be appropriate for RB.

For RB, the resample estimate can be expressed in the following form:

$$\beta^* = (X'X)^{-1}X'y^* = (X'X)^{-1}X'(\hat{y}+e^*) = (X'X)^{-1}X'(X\hat{\beta}+e^*) = \hat{\beta}+(X'X)^{-1}X'e^*. \tag{6.1}$$

In the final expression, the first term stays fixed from resample to resample, and further, even in the second term, only the e^* changes as we construct various resamples. Note that we do not want to resample the design matrix, so the $(X'X)^{-1}X'$ in the second term stays fixed. Thus the computational cost of RB per new resample is essentially that of multiplying a (fixed) p-by-n with a new vector of length n. Let $A = (X'X)^{-1}X'$, and let $\mathbf{a}_i = A[,i]$, then for RB, $\beta^* = \hat{\beta} + \sum_{i=1}^n \mathbf{a}_i e_i^*$ from (6.1) where $\mathbf{a}_i$ is a p-vector of fixed weights that does not change across resamples. We can evaluate the root function $T_n(\beta^*, \hat{\beta})$ as a proxy for $T_n(\hat{\beta}, \beta)$. Thus the computational task associated with each new iteration of RB is calculating the sum $\sum_{i=1}^n \mathbf{a}_i e_i^*$, and the corresponding computational cost is $O(n)$.

We now devise a Subsampled Residual Bootstrap (SRB) as follows. We first construct a small random subset of size b from the sample residuals, let this subset be $e_b^* = \{e_{i_1}, \ldots, e_{i_b}\}$, where let $m = n/b$ be a whole number for notational convenience. We then define the subset estimate $\beta_{(b)}^* = \hat{\beta} + (X'X)^{-1}X'e_b^{**}$ where e_b^{**} is of length n and is constructed by repeatedly concatenating e_b^* (which is of length b) m times. Thus, $\tilde{e}_b^* = \{e_{i_1}, \ldots, e_{i_b}, e_{i_1}, \ldots, e_{i_b}, \ldots, e_{i_1}, \ldots, e_{i_b}\} = J'e_b^*$ where $J_{b \times mb} = (\mathbb{I}_b \ldots \mathbb{I}_b)$ is the matrix formed by appending m identity matrices row-wise. For SRB,

$$\beta_{(b)}^* = (X'X)^{-1}X'(\hat{y} + \tilde{e}_b^*) = \hat{\beta} + (X'X)^{-1}X'J'e_b^*. \tag{6.2}$$

Note that from the final expression the corresponding computational cost is $O(b)$. Thus we have a sum of b numbers where the columns $\mathbf{b}_j = \sum_{i=1}^{m} \mathbf{a}_{j+(i-1)b}$ for $j = 1, \ldots, b$ do not change across different subsets. These are fixed weights for the subset that do not change across different subsets. We can evaluate the root function $T_n(\beta^*_{(b)}, \hat{\beta})$ as a proxy for $T_n(\hat{\beta}, \beta)$. Note that this method does not involve a second level of bootstrap, unlike SDB.

The distribution of the SRB root function needs a different scaling than that for RB. To see this, let $B = (X'X)^{-1}X'J'$ and note that $Var^*(Be^*_b) = BE^*[e^*_b(e^*_b)']B'$ which, under appropriate conditions, will converge in probability to $\sigma^2 BB'$ where σ^2 is the error variance. Therefore we expect that $(BB')^{-1/2}(\beta^*_{(b)} - \hat{\beta}) \to_D N(0, \sigma^2 I)$, and therefore the SRB version of the root function is

$$T_n^{*S}(\beta^*_{(b)}, \hat{\beta}) = (BB')^{-1/2}(\beta^*_{(b)} - \hat{\beta}).$$

6.2.1 Simulation study

Consider a d-dimensional multiple linear regression model

$$y_i = \beta_1 x_{i,1} + \ldots + \beta_d x_{i,d} + e_i$$

for $i = 1, \ldots, n$. Our parameter of interest is the d-dimensional vector of slope coefficients, whose true value is $\beta = (\beta_1, ..., \beta_d) = (0, \ldots, 0)'$. We use the usual OLS estimator $\hat{\beta}$. We also want to construct a simultaneous 95% confidence region for β. Traditionally we use the F-statistic

$$T_n(\hat{\beta}, \beta) = \frac{||X((\hat{\beta} - \beta)||^2/d}{||y - X\hat{\beta}||^2/n - d - 1}.$$

to construct the joint confidence region. Let $q_{0.95}$ be the 95% quantile of the true (unknown) distribution of $T_n(\hat{\beta}, \beta)$. Then the confidence region is given by $\{\beta : T_n(\hat{\beta}, \beta) \leq q_{0.95}\}$. In general the true distribution of T_n, and hence its quantile $q_{0.95}$, is unknown. Asymptotically, $dT_n \overset{D}{\to} \chi^2_d$ and hence quantiles from the χ^2_d/d distribution are often used as an approximation for the unknown quantiles of T_n, we call this the Normal Approximation. Alternately, $q_{0.95}$ can be estimated by the resampling techniques described in the previous section, with $\xi(Q_n) = q_{0.95}$ where Q_n is the true distribution of T_n.

We set $n = 10^5$ and $d = 10$, with $x_{i,k} \sim Pareto(\alpha = 3)$ and generate errors independently from the $\chi_1^2 - 1$ distribution. We define the error rate as

$$|\frac{\hat{q}_{0.95}}{q_{0.95}} - 1|$$

where $\hat{q}$ and q represent the estimated and true quantiles of T_n, respectively. The 'true' quantile $q_{0.95}$ is obtained by a high-accuracy simulation with 10,000 iterations from the underlying distribution.

We use subset size $b = 1000$ for SRB. We allowed the competing methods to run for 5 seconds. Figure 6.2 shows the time taken (in seconds) and the time evolution of error. We consider three competing methods of approximating $q_{0.95}$, namely the normal approximation (Norm Approx), residual bootstrap (Res Boot) and subsampled residual bootstrap (SRB). We observe that SRB is much quicker than RB in obtaining an accurate estimate, and that normal approximation is less accurate than either resampling technique.

6.2.2 Computation

The root function of interest is the so-called F statistic, defined as

$$T_n(\hat{\beta}, \beta) = \frac{||X((\hat{\beta} - \beta)||^2/d}{||y - X\hat{\beta}||^2/n - d - 1}.$$

The residual bootstrap version of this root function is given by

$$T_n(\beta^*, \hat{\beta}) = \frac{||X(\beta^* - \hat{\beta})||^2/d}{||y - X\beta^*||^2/n - d - 1} = \frac{||C_1 e^*||^2/d}{(e'e + ||C_1 e^*||^2)\,/n - d - 1}$$

where $C_1 = (X'X)^{-1/2}X'$. Note that $X'X$ is positive definite (by assumption) and hence it has a unique positive definite square root, this is $(X'X)^{1/2}$ and the inverse of this matrix is $(X'X)^{-1/2}$. Thus to run R residual bootstrap iterations, we need to compute $e'e$ and C_1 once for the whole ensemble, and for each iteration we need to compute only $||C_1 e^*||^2$, which is a computation of order n. The derivation is as follows:

$$\begin{aligned} Numerator = (\beta^* - \hat{\beta})'X'X(\beta^* - \hat{\beta}) &= e^{*\prime}X(X'X)^{-1}X'X(X'X)^{-1}X'e^* \quad \text{(from (6.1))} \\ &= e^{*\prime}C_1'C_1e^*, \end{aligned}$$

$$\begin{aligned}
Denominator &= (y - X\beta^*)'(y - X\beta^*) \\
&= \Big((y - X\hat{\beta}) - X(\beta^* - \hat{\beta})\Big)' \Big((y - X\hat{\beta}) - X(\beta^* - \hat{\beta})\Big) \\
&= ||y - X\hat{\beta}||^2 + ||X(\beta^* - \hat{\beta})||^2 - 2(y - X\hat{\beta})'X(\beta^* - \hat{\beta}) \\
&= e'e + ||C_1 e^*||^2.
\end{aligned}$$

Details for the last line:

$$\begin{aligned}
(y - X\hat{\beta})'X &= (y - X(X'X)^{-1}X'y)'X = \{(\mathbb{I} - X(X'X)^{-1}X')y\}'X \\
&= y'(\mathbb{I} - X(X'X)^{-1}X')'X = y'(\mathbb{I} - X(X'X)^{-1}X')X \\
&= y'(X - X(X'X)^{-1}X'X) = \mathbf{0}.
\end{aligned}$$

Note that for residual bootstrap, $||X(\beta^* - \hat{\beta})||^2 = ||C_1 e^*||^2$ is a asymptotically pivotal quantity for residual bootstrap, since $\text{Var}^*(C_1 e^*) = C_1 \text{Var}^*(e^*) C_1' \to \sigma^2 C_1 \mathbb{I}_n C_1' = \sigma^2 \mathbb{I}_d$. But for subsampled residual bootstrap, the analogous quantity $||X(\beta^*_{(b)} - \hat{\beta})||^2 = ||C_1 J' e^*_b||^2$ is not asymptotically pivotal since the residual resample is generated from a small subsample, which implies $\text{Var}^*(C_1 J' e^*_b) = C_1 J' \text{Var}^*(e^*_b) J C_1' \to \sigma^2 C_1 J' \mathbb{I}_b J C_1' = \sigma^2 C_1 J' J C_1' \neq \sigma^2 \mathbb{I}_d$. However the 'normalized' version $C_2 e^*_b := (C_1 J' J C_1')^{-1/2} C_1 J' e^*_b$ is asymptotically pivotal, since $\text{Var}^*(C_2 e^*_b) = C_2 \text{Var}^*(e^*_b) C_2' \to \sigma^2 C_2 \mathbb{I}_b C_2' = \sigma^2 \mathbb{I}_d$. Therefore, the SRB version of the root function is given by

$$\frac{||C_2 e^*_b||^2/d}{\left(e'e + ||C_2 e^*_b||^2\right)/n - d - 1}$$

where $C_2 = (C_1 J' J C_1')^{-1/2} C_1 J'$. Thus to run R subsampled residual bootstrap iterations, we need to compute $e'e$ and C_2 once for the whole ensemble, and for each iteration we need to compute only $||C_2 e^*_b||^2$, which is a computation of order b.

6.3 Future directions

In future work, we plan to study the theoretical properties of the ALB and the SRB. We hope to demonstrate that both these methods can consistently estimate the precision measure, and also derive higher order properties. An interesting problem is to compare the higher order properties of ALB, BLB, and SDB, which will enable us to determine useful practical guidelines on the relative accuracy

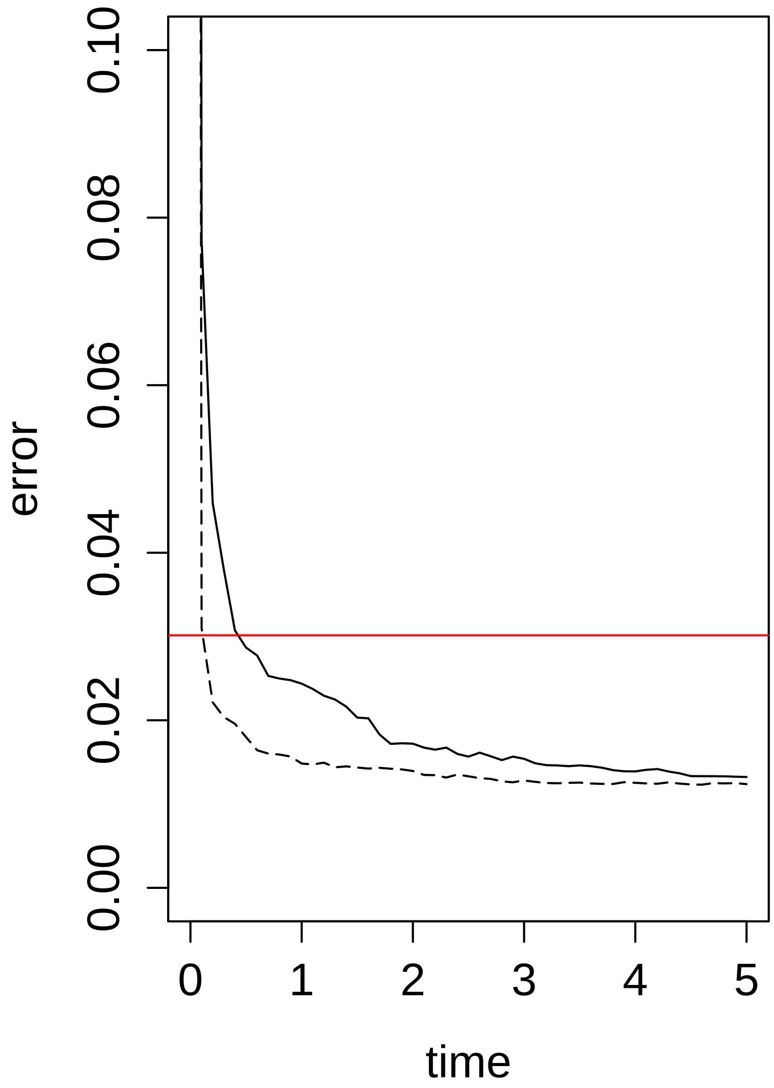


Figure 6.2: Time evolution of error rates. The red bar is the Normal Approximation error, while the errors from residual bootstrap are in solid lines and errors from SRB are in dashed lines.

these methods under various inference paradigms.

Traditionally, bootstrap methods are evaluated in terms of statistical accuracy. In the context of big data, however, the computational aspect is crucially important as well and we should evaluate methods on both computing cost and statistical accuracy. We plan to investigate this cost versus accuracy trade-off for classical residual bootstrap and subsampled residual bootstrap.

REFERENCES

Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, pages 36–43. ACM.

Ahlgren, N. and Antell, J. (2008). Bootstrap and fast double bootstrap tests of cointegration rank with financial time series. *Computational Statistics and Data Analysis*, 52(10):4754–4767.

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

Andrews, D. W. and Monahan, J. C. (1992). An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica*, 60(4):953–966.

Arcones, M. A. and Yu, B. (1994). Central limit theorems for empirical anduprocesses of stationary mixing sequences. *Journal of Theoretical Probability*, 7(1):47–71.

Beran, R. (1988). Prepivoting test statistics: a bootstrap view of asymptotic refinements. *Journal of the American Statistical Association*, 83(403):687–697.

Berkes, I., Gabrys, R., Horváth, L., and Kokoszka, P. (2009). Detecting changes in the mean of functional observations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):927–946.

Bhattacharya, R. N. and Rao, R. R. (1986). *Normal Approximation and Asymptotic Expansions.* Krieger Melbourne, FL.

Bickel, P., Götze, F., and van Zwet, W. (1997). Resampling fewer than n observations: Gains, losses, and remedies for losses. *Statistica Sinica*, 7:1–31.

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

Bickel, P. J. and Sarkar, P. (2015). Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society: Series B*, doi: 10.1111/rssb.12117.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.

Bühlmann, P. (1994). Blockwise bootstrapped empirical process for stationary sequences. *Annals of Statistics*, 22:995–1012.

Bühlmann, P. (1995). The blockwise bootstrap for general empirical processes of stationary sequences. *Stochastic Processes and their Applications*, 58(2):247–265.

Bühlmann, P. and Künsch, H. R. (1999). Block length selection in the bootstrap for time series. *Computational Statistics and Data Analysis*, 31(3):295–310.

Chang, J. and Hall, P. (2014). Double-bootstrap methods that use a single double-bootstrap simulation. *arXiv preprint arXiv:1408.6327*.

Chaudhuri, K., Chung, F., and Tsiatas, A. (2012). Spectral clustering of graphs with general degrees in the extended planted partition model. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 23:35.1–35.23.

Cover, T. M. and Thomas, J. A. (2012). *Elements of Information Theory*. John Wiley & Sons.

d'Ancona, M. (2015). Could London's next mayor really be another Old Etonian? *The Guardian*, 07-19-2015.

Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, 18:173–183.

Davidson, R. and MacKinnon, J. G. (2000). Improving the reliability of bootstrap tests. Queens University Working paper no. 995.

Davidson, R. and MacKinnon, J. G. (2002). Fast double bootstrap tests of nonnested linear regression models. *Econometric Reviews*, 21(4):419–429.

Davidson, R. and MacKinnon, J. G. (2007). Improving the reliability of bootstrap tests with the fast double bootstrap. *Computational Statistics and Data Analysis*, 51(7):3259–3281.

Dudley, R. M. (1999). *Uniform Central Limit Theorems*, volume 23. Cambridge University Press.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.

Erdös, P. and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297.

Fienberg, S. E., Meyer, M. M., and Wasserman, S. S. (1985). Statistical analysis of multiple sociometric relations. *Journal of the American Statistical Association*, 80:51–67.

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

Gao, J., Liang, F., Fan, W., Sun, Y., and Han, J. (2009). Graph-based consensus maximization among multiple supervised and unsupervised models. *Advances in Neural Information Processing Systems*, 22:585–593.

Garcia-Soidan, P. H. and Hall, P. (1997). On sample reuse methods for spatial data. *Biometrics*, 53:273–281.

Gavin, A.-C., Bösche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A.-M., Cruciat, C.-M., Remor, M., Höfert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., Gnau, V., Bauch, A., Bastuck, S., Huhse, B., Leutwein, C., Heurtier, M.-A., Copley, R. R., Edelmann, A., Querfurth, E., Rybin, V., Drewes, G., Raida, M., Bouwmeester, T., Bork, P., Seraphin, B., Kuster, B., Neubauer, G., and Superti-Furga, G. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147.

Giacomini, R., Politis, D. N., and White, H. (2013). A warp-speed method for conducting monte carlo experiments involving bootstrap estimators. *Econometric Theory*, 29(3):567–589.

Giné, E. and Zinn, J. (1984). Some limit theorems for empirical processes. *The Annals of Probability*, 12(4):929–989.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

Goldenberg, A., Zheng, A., Fienberg, S., and Airoldi, E. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2:129–233.

Greene, D. and Cunningham, P. (2013). Producing a unified graph representation from multiple social network views. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 118–121. ACM.

Hall, P. (1985). Resampling a coverage pattern. *Stochastic Processes and their Applications*, 20:231–246.

Hall, P., Horowitz, J. L., and Jing, B.-Y. (1995). On blocking rules for the bootstrap with dependent data. *Biometrika*, 82(3):561–574.

Holland, P., Laskey, K., and Leinhardt, S. (1983). Stochastic blockmodels: first steps. *Social Networks*, 5:109–137.

Huberman, B. A. and Adamic, L. A. (1999). Internet: growth dynamics of the World-Wide Web. *Nature*, 401:131.

Ji, M., Sun, Y., Danilevsky, M., Han, J., and Gao, J. (2010). Graph regularized transductive classification on heterogeneous information networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 570–586. Springer.

Jin, J. (2012). Fast network community detection by SCORE. *arXiv preprint arXiv:1211.5803*.

Jonsson, P. F., Cavanna, T., Zicha, D., and Bates, P. A. (2006). Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. *BMC Bioinformatics*, 7(1):2.

Jordan, M. I. (2013). On statistics, computation and scalability. *Bernoulli*, 19(4):1378–1390.

Joseph, A. and Yu, B. (2013). Impact of regularization on spectral clustering. *arXiv preprint arXiv:1312.1733*.

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

Kiefer, N. M. and Vogelsang, T. J. (2005). A new asymptotic theory for heteroskedasticity-autocorrelation robust tests. *Econometric Theory*, 21:1130–1164.

Kiefer, N. M., Vogelsang, T. J., and Bunzel, H. (2000). Simple robust testing of regression hypotheses. *Econometrica*, 68(3):695–714.

Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. (2014). A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76:795–816.

Kley, T., Volgushev, S., Dette, H., and Hallin, M. (2014). Quantile spectral processes: Asymptotic analysis and inference. *arXiv preprint arXiv:1401.8104*.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. Springer.

Kosorok, M. R. (2008). *Introduction to Empirical Processes and Semiparametric Inference*. Springer, New York.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics*, 17(3):1217–1241.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3):1217–1241.

Lahiri, S. N., , and Zhu, J. (2006). Resampling methods for spatial regression models under a class of stochastic designs. *Annals of Statistics*, 34:1774–1813.

Lahiri, S. N. (1999). Asymptotic distribution of the empirical spatial cumulative distribution function predictor and prediction bands based on a subsampling method. *Probability Theory and Related Fields*, 114:55–84.

Lahiri, S. N. (2003a). Springer, Resampling Methods For Dependent Data.

Lahiri, S. N. (2003b). Central limit theorems for weighted sums of a spatial process under a class of stochastic and fixed designs. *Sankhyā*, 65:356–388.

Lahiri, S. N., Kaiser, M. S., Cressie, N., and j. Hsu, N. (1999). Prediction of spatial cumulative distribution functions using subsampling. *Journal of the American Statistical Association*, 94:86–110.

Lahiri, S. N. and Mukherjee, K. (2004). Asymptotic distributions of m-estimators in a spatial regression model under some fixed and stochastic spatial sampling designs. *Annals of the Institute of Statistical Mathematics*, 56:225–250.

Laptev, N., Zaniolo, C., and Lu, T.-C. (2012). BOOT-TS: A Scalable Bootstrap for Massive Time-Series Data. `http://cs.ucla.edu/~zaniolo/papers/biglearning2012_submission_3.pdf`. [Online; accessed 13-December-2014].

Le, C. M., Levina, E., and Vershynin, R. (2015). Optimization via low-rank approximation for community detection in networks. *arXiv preprint arXiv:1406.0067*.

Liu, R. Y. and Singh, K. (1992). Moving blocks jackknife and bootstrap capture weak dependence. *Exploring the limits of bootstrap*, 225:248.

Lorrain, F. and White, H. C. (1971). Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1:49–80.

Milgram, S. (1967). The small world problem. *Psychology Today*, 2(1):60–67.

Naik-Nimbalkar, U. V. and Rajarshi, M. B. (1994). Validity of blockwise bootstrap for empirical processes with stationary observations. *Annals of Statistics*, 22:980–994.

Newman, M. E. J. (2010). *Networks: An Introduction.* Oxford University Press.

Nordman, D. J. and Lahiri, S. N. (2004). On optimal spatial subsample size for variance estimation. *Annals of Statistics*, 32:1981–2027.

Peligrad, M. (1998). On the blockwise bootstrap for empirical processes for stationary sequences. *Annals of Probability*, 26:877–901.

Politis, D. N., Paparoditis, E., and Romano, J. P. (1998). Large sample inference for irregularly spaced dependent observations based on subsampling. *Sankhyā*, 60:274–292.

Politis, D. N. and Romano, J. P. (1993). Nonparametric resampling for homogeneous strong mixing random fields. *Journal of Multivariate Analysis*, 47:301–328.

Politis, D. N. and Romano, J. P. (1994a). Large sample confidence regions based on subsamples under minimal assumptions. *The Annals of Statistics*, 22(4):2031–2050.

Politis, D. N. and Romano, J. P. (1994b). The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313.

Politis, D. N. and Sherman, M. (2001). Moment estimation for statistics from marked point processes. *Journal of the Royal Statistical Society: Series B*, 63:261–275.

Qin, T. and Rohe, K. (2013). Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems*, pages 3120–3128.

Radulović, D. (1996). The bootstrap for empirical processes based on stationary observations. *Stochastic processes and their applications*, 65(2):259–279.

Radulović, D. (2002). On the bootstrap and empirical processes for dependent sequences. In *Empirical process techniques for dependent data*, pages 345–364. Springer.

Radulović, D. (2009). Another look at the disjoint blocks bootstrap. *Test*, 18(1):195–212.

Rao, C. R. and Zhao, L. C. (1992). Approximation to the distribution of m-estimates in linear models by randomly weighted bootstrap. *Sankhya: The Indian Journal of Statistics*, 54:323–331.

Rho, Y. and Shao, X. (2013). Improving the bandwidth-free inference methods by prewhitening. *Journal of Statistical Planning and Inference*, 143(11):1912–1922.

Richard, P. (2009). Modified fast double sieve bootstraps for adf tests. *Computational Statistics & Data Analysis*, 53(12):4490–4499.

Rohe, K., Chatterjee, S., and Yu, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915.

Rubin, D. B. (1981). The Bayesian bootstrap. *Annals of Statistics*, 9:130–134.

Shao, J. and Tu, D. (1995). *The Jackknife and Bootstrap.* Springer-Verlag, New York.

Shao, X. (2009). Extended tapered block bootstrap. *Statistica Sinica*, 20:807–821.

Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association*, 105:218–235.

Shao, X. and Politis, D. N. (2013). Fixed-b subsampling and block bootstrap: improved confidence sets based on p-value calibration. *Journal of the Royal Statistical Society: Series B*, 75:161–184.

Sherman, M. (1996). Variance estimation for statistics computed from spatial lattice data. *Journal of the Royal Statistical Society: Series B*, 58:509–523.

Sherman, M. and Carlstein, E. (1994). Nonparametric estimation of the moments of a general statistic computed from spatial data. *Journal of the American Statistical Association*, 89:496–500.

Sullivan, A. (2009). Forbes' definition of "liberal". *The Atlantic*, 01-24-2009.

Sun, Y. and Han, J. (2012). *Mining Heterogeneous Information Networks: Principles and Methodologies.* Morgan & Claypool Publishers.

van der Vaart, A. W. and Wellner, J. A. (1996). *Weak Convergence and Empirical Processes - Springer Series in Statistics.* Springer, New York.

Varadarajan, T., Eaves, E., and Alberts, H. R. (2009). The 25 most influential liberals in the US media. *Forbes*, 01-22-2009.

Volgushev, S. and Shao, X. (2014). A general approach to the joint asymptotic analysis of statistics from sub-samples. *Electronic Journal of Statistics*, 8:390–431.

Wang, Y. H. (1993). On the number of successes in independent trials. *Statistica Sinica*, 3:295–312.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442.

White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5):1097–1126.

Zhang, X., Shao, X., Hayhoe, K., and Wuebbles, D. J. (2011). Testing the structural stability of temporally dependent functional observations and application to climate projections. *Electronic Journal of Statistics*, 5:1765–1796.

Zhao, Y., Levina, E., and Zhu, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40:2266–2292.