Methods for High Dimensional Analysis, Multiple Testing, and Visual Exploration

by

Aaron J. Fisher

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland

September, 2016

Abstract

My thesis work focuses on aiding the practical implementation of advanced statistical methods. Chapter 2 concerns the common practice of visual exploratory data analysis, and the extent to which humans can visually detect statistical significance from plots. We find that human accuracy in detecting significance was initially poor, but improved with practice. Chapter 3 aids the implementation of bootstrap principal component analysis, by providing significant computational improvements. In a dataset of brain magnetic resonance images, the proposed method can reduce bootstrap standard error computation times from approximately 4 days to 47 minutes. Chapter 4 proposes an approximate optimization technique for adaptive clinical trials, aimed at lowering the expected sample size or expected duration of a trial.

Dedication

To my family and friends,

Thank you to my great friends in and outside of Baltimore, for helping to make this city a home. Thank you to my parents, Ellen and Reuben, and my brother, David, for your constant love and support. I am beyond lucky to have you in my life. Lots of love,

-Aaron

Thesis Committee

Primary Readers

Vadim Zipunnikov (Primary Advisor)
Assistant Professor
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Michael Rosenblum
Associate Professor
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Adam Spira
Associate Professor
Department of Mental Health
Johns Hopkins Bloomberg School of Public Health

Kathleen Zackowski
Associate Professor
Kennedy Krieger Institute, and
Departments of Physical Medicine & Rehabilitation, and Neurology at
Johns Hopkins University School of Medicine

Alternate Readers

Brian Caffo (Co-Advisor)
Professor
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Jennifer Schrack
Assistant Professor
Department of Epidemiology
Johns Hopkins Bloomberg School of Public Health

Acknowledgments

My research has been largely supported by the National Institute of Environmental Health Sciences (grant T32ES012871). The methodological research in Chapter 3 was additionally supported by the National Institute of Biomedical Imaging And Bioengineering (grants R01 EB012547 and P41 EB015909), the National Institute of Neurological Disorders and Stroke (grant R01 NS060910), and the National Heart, Lung, and Blood institute (grant R01 HL123407). Recording and maintenance of the MRI dataset used in Chapter 3 was supported by the National Institute on Aging (grant R01 AG10785). The research in Chapter 4 was supported primarily by the Participant-Centered Outcomes Research Institute (ME-1306-03198) and the U.S. Food and Drug Administration (HHSF223201400113C). Additional information on funding support is provided at the end of each chapter.

I am deeply appreciative for the time and attention of my thesis committee, and for the advice they have given regarding this research.

In particular, thank you to my supportive and talented advisors, Vadim Zipunnikov and Brian Caffo. I have enjoyed working with you tremendously.

Thank you to Michael Rosenblum for the support, guidance, and knowledge you have given me throughout my time here. Thank you also to Ciprian Crainiceanu for your encouragement and mentorship, and to Roger Peng and Tom Louis, for your guidance, and for leading the Environmental Biostatistics and Epidemiology Working Group. I am also thankful to have worked with many other outstanding statisticians in the department during my thesis work, including Yates Coley, Brooke G Anderson, Jeff Leek, and Scott Zeger.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

My thesis work focuses on aiding the practical implementation of advanced statistical methods. This goal is achieved by improving their computational speed, increasing their efficiency, or providing insight into their practical use. A common component across the research presented here is an attention to multiple hypothesis testing problems. In Chapter 2, we study the extent to which multiple informal significance tests during a visual exploratory data analysis may bias later hypothesis tests towards significance. In Chapter 3, we discuss high dimensional confidence regions that simultaneously test several aspects of a multivariate test statistic. In Chapter 4, we compare approaches for repeatedly testing a set of hypotheses as data accrues. While these chapters all pertain to multiple testing problems, they also span a wide range of applications and primary goals.

Chapter 2 generally discusses the variability in statistical conclusions due to differences in how statistical methods are implemented by analysts. This focus on human analyst variability stands in contrast to traditional statistics

research on uncertainty due to random sampling. Understanding which statistical methods are most easily replicated across analysts is an important aspect of creating guidelines and recommendations for analysis. We look here at the specific practice of visually observing scatterplots in order to identify significant relationships between variables – a common exploratory procedure that is highly subject to human variability. We find that analysts have poor baseline accuracy in visually identifying significant relationships, but that accuracy in certain scenarios can improve with practice.

Chapter 3 proposes a novel, fast algorithm for estimating sampling variability in patterns found in large datasets. We specifically look at patterns discovered by principal component analysis (PCA), a common tool for summarizing variability in high dimensional data (e.g. in brain imaging, genomics, or air pollutant compositions). We develop a fast, exact bootstrap algorithm for estimating standard errors and confidence regions for PCA outputs, or for methods that depend on PCA outputs. In a dataset of brain magnetic resonance images, we demonstrate that our method can reduce standard error computation times from approximately 4 days to 47 minutes.

Chapter 4 proposes an approximate optimization procedure for reducing the expected sample size of an adaptive clinical trial. We consider a class of adaptive trials known as adaptive enrichment designs, which allow the enrollment criteria to be modified at interim analyses based on preset decision rules. The trial design also includes parameters that characterize the way in which multiplicity corrections are done for tests of treatment effects in each subpopulation. An obstacle to using these designs is that there is no general approach to determine what decision rules and other design parameters will lead to good performance

for a given research problem. To address this, we present a simulated annealing approach for optimizing the parameters of an adaptive enrichment design for a given scientific application. Optimization is done with respect to either expected sample size or expected trial duration, and subject to constraints on power and Type I error rate. We find that optimized designs can be substantially more efficient than simpler designs using Pocock or O'Brien-Fleming boundaries. Much of this added benefit comes from optimizing the decision rules concerning when to stop a subpopulation's enrollment, or the entire trial, due to futility.

Chapter 2

A randomized trial in a massive online open course shows people don't know what a statistically significant relationship looks like, but they can learn

2.1 Introduction

Over the last two decades there has been a dramatic increase in the amount and variety of data available to scientists, physicians, and business leaders in nearly every area of application. Statistical literacy is now critical for anyone consuming data analysis reports, including scientific papers, newspaper reports Beyth-Marom et al. (2008), legal cases Gastwirth (1988), and medical test results Schwartz et al. (1997); Sheridan et al. (2003). A lack of sufficient training in statistics and data analysis has been responsible for the retraction of high-profile papers Ledford (2011), the cancellation of clinical trials Pelley (2012), and mistakes in papers used to justify major economic policy initiatives Cassidy (2013).

Despite the critical importance of statistics and data analysis in modern life, we have relatively little empirical evidence about how statistical tools work in the hands of typical analysts and consumers. The most well-studied statistical tool is the visual display of quantitative information. Previous studies have shown that humans have difficulty interpreting linear measures of correlation Cleveland et al. (1982), are better at judging relative positions than relative angles Heer and Bostock (2010); Cleveland et al. (1985), and view correlations differently when plotted on different scales Cleveland et al. (1982). These studies show that mathematically correct statistical procedures may have unintended consequences in the hands of users. The real effect of a statistical procedure depends, to a large extent, on psychology and cognitive function.

Here we perform a large-scale study of the ability of average data analysts to detect statistically significant relationships from scatterplots. Our study compares two of the most common data analysis tasks, making scatterplots and calculating P-values. It has been estimated that as many as 80% of the plots published across all scientific disciplines are scatterplots Tufte and Graves-Morris (1983). At the same time, and despite widely publicized controversy over their use Nuzzo (2014), P-values remain the most common choice for reporting a statistical summary of the relationship between two variables in the scientific literature. In the decade 2000-2010, 15,653 P-values were reported in the abstracts of the *The Lancet*, *The Journal of the American Medical Association*, *The New England Journal of Medicine*, *The British Medical Journal*, and *The American Journal of Epidemiology* Jager and Leek (2007).

Data analysts frequently use exploratory scatterplots for model selection and

building. Selecting which variables to include in a model can be viewed as visual hypothesis testing where the test statistic is the plot and the measure of significance is human judgement. However, it is not well known how accurately humans can visually classify significance when looking at graphs of raw data. This classification task depends on both understanding what combinations of sample size and effect size constitute significant relationships, and being able to visually distinguish these effect sizes. We performed a set of experiments to (1) estimate the baseline accuracy with which subjects could visually determine if two variables showed a statistically significant relationship; (2) test whether accuracy in visually classifying significance was changed by the number of data points in the plot or the way the plot was presented; and (3) test whether accuracy in visually classifying significance improved with practice. Our intuition is that potential improvements with practice would be better explained by an improved cognitive understanding of statistical significance, rather than an improved perceptive ability to distinguish effect sizes.

2.2 Methods and results

Our study was conducted within the infrastructure of a statistics massive online open course (MOOC). While MOOCs have previously been used to study MOOCs Do et al. (2013); Mak et al. (2010); Liyanagunawardena et al. (2013), to our knowledge this is the first example of a MOOC being used to study the practice of science. Specifically, our survey was conducted as an ungraded, voluntary exercise within the Spring 2013 Data Analysis Coursera class. This class was 8 weeks long, and consisted of lecture content, readings, and a weekly quiz.

Although 121,257 students registered for the course, only 5,306 completed the final weekly quiz. The survey was made available to all students in the class, and 2,039 students responded – approximately 38% relative to the number of active users. In one of the weekly quizzes preceding the survey, students were asked two questions relating to the concept of P-values (see supplemental materials for specific question text). Students had two attempts at each question and their accuracies were: 73.5% (1st attempt, 1st question), 96.1% (2nd attempt, 1st question), 73.1% (1st attempt, 2nd question), and 95.4% (2nd attempt, 2nd question). These questions were not identical to the questions in our survey but suggest that students understand the concepts behind a P-value, assuming that almost all students completed the graded quizzes before submitting responses to the optional exercises that followed.

Each student who participated in the survey was shown a set of bi-variate scatterplots (examples shown in Figure 2.1). The set of plots included eight plots from seven different categories (Table 2.1), with two plots from the reference category (of which one was significant and one was not) and one plot from each of the other categories (each randomly chosen to be either significant or non-significant). These plot categories (Table 2.1) were selected to allow analysis of whether students' accuracy in visually classifying significance changed based on the number of data points in the plot, or the plot's presentation style. Each set of plots shown to a user was randomly selected from a library containing 10 plots from each category (see supplemental materials for full library and generating code), of which half were statistically significant (P-values from testing the slope coefficient in a linear regression relating X and Y were between 0.023 and 0.025; e.g., Figure 2.1 left) and half were not statistically significant (P-values between

0.33 and 0.35; e.g., Figure 2.1, right).

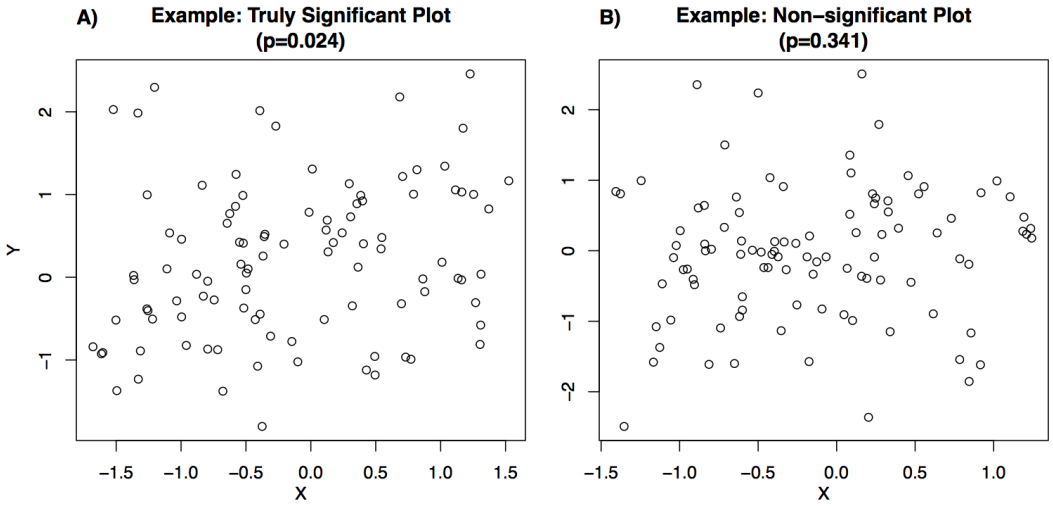

Figure 2.1: Examples of Plots Shown to Users

For each plot, students were asked to visually determine whether the bivariate relationship shown was statistically significant at the 0.05 level (in the example plots shown in Figure 2.1, the correct answer would have been "statistically significant" for the left plot, for which the P-value of a linear relationship between the X and Y variables is 0.024, and "not statistically significant" for the right plot, for which the P-value is 0.341). All eight plots were shown at the same time and students submitted responses for all plots in a single submission. Students were also able to submit a partial response by leaving some of the survey questions blank. 94.4% of users completed their first attempt of the survey. After submitting their responses, students were shown the correct answers and given the opportunity to retake the survey with a new set of plots. Students were not told any information about the structure of the survey and so were not able to use the structure of the survey (e.g., the fact that one of the

Reference	100 data points (e.g., Figure 2.1)
Smaller n	35 data points
Larger n	200 data points
Best-fit line	100 data points, with best fit line added
Lowess	100 data points, with smooth lowess curve added (using R "lowess" function)
Axis Scale	100 data points, with the axis range increased to 1.5 standard deviations outside X and Y variable ranges (e.g., "zoomed out") Cleveland et al. (1982)
Axis Label	100 data points, with fictional X- and Y-axis labels added corresponding to activation in a brain region (e.g., "Cranial Electrode 33 (Standardized)" versus "Cranial Electrode 92 (Standardized)")

Table 2.1: Plot Categories Shown to Users

"Reference" plots was significant and one was not) to improve their accuracy.

To analyze responses, we created separate models for the probability of correctly visually classifying significance in: (1) graphs that showed two variables with a statistically significant relationship (e.g., Figure 2.1, left) and (2) graphs that showed two variables with a statistically non-significant relationship (e.g., Figure 2.1, right). These two types of visual classification correspond to the separate accuracy metrics: human sensitivity to significance (accuracy in giving a positive result in cases where a condition is true) and human specificity to non-significance (accuracy in giving a negative result in cases where a condition is false). In this framework, the hypothetical baseline case where humans have no ability to classify significance corresponds to the sensitivity rate being equal to one minus the specificity rate, which means that the probability of visually classifying a plot as significant is unaffected by the actual significance level of the plot. Accuracy in both metrics was modeled by logistic regressions with person-specific random intercept terms, using the "lme4" package in R (see supplemental materials).

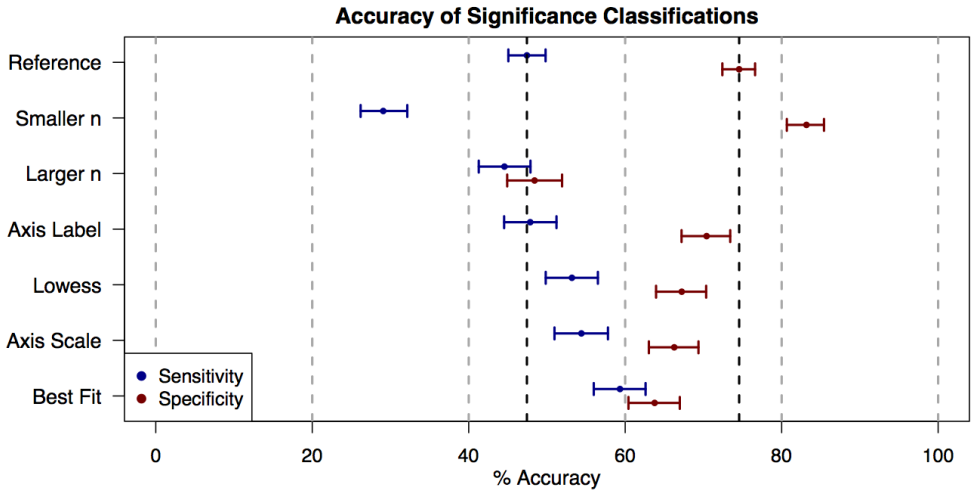

Figure 2.2: Accuracy of Significance Classifications Under Different Conditions: Point estimates and confidence intervals for classification accuracy for each presentation style (Table 2.1). Accuracy rates for plots with truly significant underlying relationships (sensitivity) are shown in blue, and accuracy rates for plots with non-significant underlying relationships (specificity) are shown in red.

We found that, overall, subjects tended to be conservative in their classifications of significance. In the reference category (100 data points; Table 2.1, examples in Figure 2.1), students accurately classified graphs of significant relationships as significant only 47.4% (95% CI: 45.1%-49.7%) of the time (i.e., 47.4% sensitivity) and accurately classified graphs of non-significant relationships as non-significant 74.6% (95% CI: 72.5%-76.6%) of the time (i.e., 74.6% specificity) (Figure 2.2). Specificity exceeded sensitivity across all of the plot categories presented (Figure 2.2).

When comparing the reference plot category of 100 data points to other plot categories (Table 2.1), sensitivity and specificity were in some cases significantly

changed by the number of points displayed in the graph or the style of graph presentation (Figure 2.2). Changes to the plots that increased sensitivity correlated with changes that decreased specificity. For example, reducing the number of data points shown ("Smaller n" plot category) significantly decreased sensitivity (Odds Ratio (OR) = 0.454, 95%CI: 0.385-0.535) and increased specificity (OR = 1.67, 95%CI: 1.39-2.04). Adding visual aids (best-fit line, lowess curve) significantly improved sensitivity (OR = 1.62 and 1.26 respectively, with 95%CIs: 1.38-1.89 and 1.08-1.47), but significantly reduced specificity (OR = 0.600 and 0.699 respectively, with 95%CIs: 0.508-0.709 and 0.590-0.829). Changing the scale of the axes also increased sensitivity (OR = 1.32, 95%CI: 1.13-1.55), but decreased specificity (OR = 0.670, 95%CI: 0.567-0.792). Finally, changing the axes label had no significant effect on sensitivity (OR = 1.02, 95%CI: 0.871-1.19) and only a marginally significant effect on specificity (OR = 0.811, 95%CI: 0.682-0.965). Because any gain in either specificity or sensitivity tended to come at the cost of the other, none of these plot categories represented a uniform increase in accuracy across all true significance levels of the data underlying the plots.

The exception to this counter-balancing trend came in "Larger n" plots of 200 data points, where students showed a significant drop in specificity (OR = 0.320, 95%CI: 0.271-0.377), and no significant change in sensitivity (OR = 0.891, 95%CI: 0.763-1.04). For plots in this category, the probability that users would classify a relationship as significant was fairly similar across truly significant plots and nonsignificant plots. One possible explanation for this is that larger samples require a lower correlation to attain the same significance level. If the correlation becomes imperceptibly small, then the probability that an

observer classifies a relationship as significant might be less affected by the true significance level of the plot.

To test if accuracy in visually classifying significance improved with practice, we selected only the students who submitted the quiz multiple times (101 students) and compared accuracy rates between these students' first and second attempts. Of these students, 92% completed their first attempt of the survey, and 99% completed their second attempt of the survey. Because these students self-selected to take the survey twice, they may not form a representative sample of the broader population. However, they may still be representative of motivated students who wish to improve their statistical skills.

We found that, for the "Reference", "Best Fit", and "Smaller n" categories, sensitivity improved significantly on the second attempt of the survey (OR = 5.27, 2.98, and 4.51, with 95%CIs: 2.69-10.33, 1.28-6.92, and 1.79-11.37; Figure 2.3). For the "Reference" and "Best Fit" categories, the sensitivity improvements were not associated with significant changes in specificity, indicating an improvement in overall accuracy in the visual classification of significance. In the "Smaller n" plot category however, the increased sensitivity came at the cost of a significant decrease in specificity (OR = 0.163, 95%CI: 0.059-0.447). For plot in this "Smaller n" category, practice did not necessarily improve overall accuracy in visually classifying significance, but rather increased a student's odds of classifying any graph as "significant," regardless of whether the relationship it displayed was truly significant. It is possible that this was due to students over-correcting for their conservatism on their first attempts of the survey. For the remaining plot categories ("Larger n", "Axis Label", "Lowess", "Axis Scale"), there were no statistically significant changes in sensitivity or

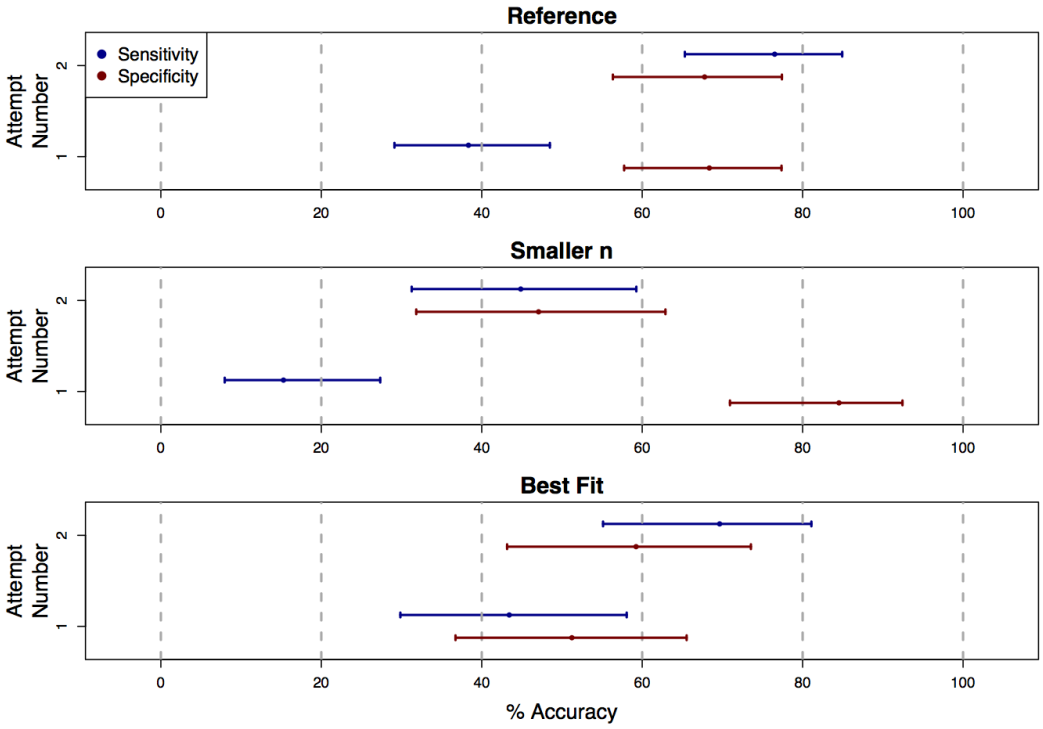

Figure 2.3: Classification Accuracy on Repeat Attempts of the Survey: Each plot shows point estimates and confidence intervals for accuracy rates of human visual classifications of statistical significance on the first and second attempt of the survey. For the truly significant underlying P-values, users showed a significant increase in accuracy (sensitivity) on the second attempt of the survey for the "Reference," "Smaller n," and "Best Fit" presentation styles. For non-significant underlying P-values, accuracy (specificity) decreased significantly for the "Smaller n" category. Because these accuracy rates were estimated only based on the data from students who submitted more than one response to the survey, the confidence intervals here are wider than those in Figure 2.2.

specificity between first and second attempts.

2.3 Discussion

Our research focuses on the question of how accurately statistical significance can be visually perceived in scatterplots of raw data. This work is a logical extension of previous studies on the visual perception of correlation in raw data scatterplots Cleveland et al. (1982); Meyer and Shinar (1992); Rensink and Baldridge (2010), and on the visual perception of plotted confidence intervals in the absence of raw data Belia et al. (2005). The results of this trial are not only relevant towards anyone who wishes to more intuitively understand P-values in scientific literature, but also towards designers and observers of scatterplots. Designers of plots should keep in mind that adding trend lines to a plot tends to make viewers more likely to perceive the underlying relationship as significant, regardless of the relationship's actual significance level, so that they can prevent their plots from misleading viewers. Similarly, viewers of scatterplots may want to slightly discount their perception of statistical significance when trend lines are shown.

Our results also suggest that, on average, readers can improve their ability to visually perceive statistical significance through practice. Our intuition is that this improvement is better explained by an improved understanding of what effect sizes constitute significant relationships, rather than an improved ability to visually distinguish these effect sizes. It would follow that the apparent baseline poor accuracy in visually detecting significance is largely due to a false intuition for what constitutes significant relationships. A broad movement

towards practicing the task of visually classifying significance could improve this intuition, and better the efficiency and clarity of communication in science.

To help readers train their sense for P-values, we've created an interactive online application where users can explore the connection between the significance level of a bi-variate relationship and how the data for that relationship appears in a scatterplot (`http://glimmer.rstudio.com/afisher/EDA/`). Users can see the visual effect of changing sample size while holding the P-value constant. They can also add lowess curves and best-fit lines to the scatterplot.

This research is also relevant to debate over the misuse of EDA. It has been argued that when EDA and formal hypothesis testing are applied to the same dataset, the "data snooping" committed through EDA process can increase the Type I error rates of the formal hypothesis tests Berk et al. (2010). However, the apparently low sensitivity with which humans can detect statistically significant relationships in scatterplots implies that both the costs of EDA misuse, as well as the benefits of responsibly conducted EDA, may be smaller than expected.

Data analysis involves the application of statistical methods. Our study highlights that even when the theoretical properties of a statistic are well understood, the actual behavior in the hands of data analysts may not be known. Our study highlights the need for placing the practice of data analysis on a firm scientific footing through experimentation. We call this idea evidence based data analysis, as it closely parallels the idea evidence based medicine, the term for scientifically studying the impact of clinical practice. Evidence based data analysis studies the practical efficacy of a broad range of statistical methods when used, sometimes imperfectly, by analysts with different levels of statistical training. Further research in evidence based data analysis may be one way

to reduce the well-documented problems with reproducibility and replicability of complicated data analyses.

Supplemental materials

Supplemental materials, including more details on our survey, and code for our analysis, are available at

`https://github.com/aaronjfisher/visual_pvalue/tree/master`

The organization of the supplement is described in the "readMe.md" file.

Acknowledgements

This chapter is joint work with G. Brooke Anderson, Roger Peng, and Jeff Leek. It appeared in PeerJ on October 16, 2014.

Chapter 3

Fast, exact bootstrap principal component analysis for $p > 1$ million

3.1 Introduction

Principal component analysis (PCA) (Jolliffe, 2005) is a dimension reduction technique that is widely used fields such as genomics, survey analysis, and image analysis. Given a multidimensional dataset, PCA identifies the set of basis vectors such that the sample subjects' projections onto these basis vectors are maximally variable. These new basis vectors are called the sample principal components (PCs), and the subjects' coordinates with respect to these basis vectors are called the sample scores. The sample PCs can be thought of as estimates of the population PCs, or the eigenvectors of the population covariance matrix. It has been shown that, as dimension increases, whether or not the sample PCs converge to their population counterparts depends on the rate of sample size growth, the rate of dimension growth, and the spacing of the eigenvalues of the population covariance matrix (Shen et al. (2012a), for a recent literature review, see Koch (2013)). Nadler (2008) and Shen et al. (2012a)

discuss PC consistency under the "spike covariance" model introduced by Johnstone (2001), where the first several eigenvalues of the population covariance matrix are assumed to be much larger than the remaining eigenvalues. Jung and Marron (2009) introduced consistency conditions for cases where sample size is fixed, dimension grows, and groups of eigenvalues grow with dimension at different rates. Shen et al. (2013) discuss consistency conditions for sparse PCA, when the first eigenvector of the population covariance matrix can be assumed sparse. Consistency conditions for the n-length, right singular vectors of high dimensional sample data matrices are discussed by Leek (2011) and Shen et al. (2012b).

A fundamental drawback of the PCA algorithm is that it is purely descriptive – there is no clear method for estimating the sampling variability of the scores, the PCs, or proportion of variance that each PC explains. Analytically derived, asymptotic confidence intervals for PCs typically require the assumption of normally distributed data (Girshick, 1939; Tipping and Bishop, 1999), or existence and computation of fourth order moments which results in $O(p^4)$ complexity (Kollo and Neudecker, 1993, 1997; Ogasawara, 2002), where p is the sample dimension. As an alternative to analytical, asymptotic confidence intervals, Diaconis and Efron (1983) proposed bootstrap based confidence intervals for PCA results. Hall and Hosseini-Nasab (2006) gave a theoretical justification for using bootstrap confidence regions to estimating sampling variability of functional PCA output. Goldsmith et al. (2013) applied a bootstrap procedure in functional PCA to estimate confidence bands for subject-level underlying functions, accounting for additional uncertainty coming from the PC decomposition. Salibián-Barrera et al. (2006) use the bootstrap in the context of a

robust PCA procedure. There, the authors applied an eigenvalue decomposition to a robust estimate of the population shape matrix, which is a scaled version of the population covariance matrix. The bootstrap has also been discussed in the context of factor analysis (Chatterjee, 1984; Thompson, 1988; Lambert et al., 1991), and in the context of determining the number of nontrivial components in a dataset (Lambert et al., 1990; Jackson, 1993; Peres-Neto et al., 2005; Hong et al., 2006). However, when applying the bootstrap to PCA in the high dimensional setting, the challenge of calculating and storing the PCs from each bootstrap sample can make the procedure computationally infeasible.

To address this computational challenge, we outline methods for exact calculation of PCA in high dimensional bootstrap samples that are an order of magnitude faster than the current standard methods. These methods leverage the fact that all bootstrap samples occupy the same n-dimensional subspace, where n is the original sample size. Importantly, this leads to bootstrap variability of the PCs being limited to rotational variability within this subspace. To improve computational efficiency, we shift operations to be computed on the low dimensional coordinates of this subspace before projecting back to the original p-dimensional space.

There has been very little work applying bootstrap to PCA in the high dimensional context, largely due to computational bottlenecks. The methods we propose drastically reduce these bottlenecks, allowing for simulation studies of PCA in high dimensions, and for further study of bootstrap PCA in real world, high dimensional scientific applications.

Our methods can also be directly applied to determine the resampling-based variability of any model that depends on a singular value decomposition of the

sample data matrix. For example, in Independent Component Analysis (ICA, Bell and Sejnowski, 1995), the first step is typically to use PCA to represent the data on a low dimensional space (Calhoun et al., 2001). Other examples include bootstrap and cross-validation variability for principal component regression (PCR), ridge regression, and, more generally, regression with quadratic penalties.

The remainder of this paper is organized as follows. Section 3.1.1 presents some initial mathematical notation, and gives a basic summary of PCA and the bootstrap procedure. Section 3.1.2 outlines the intuition for fast bootstrap PCA. Section 3.2 discusses two motivating data examples – one based on sleep electroencephalogram (EEG) recordings, and one based on brain magnetic resonance images (MRIs). Section 3.3 presents the full details of our methods for fast, exact bootstrap PCA. The computation complexity of our methods depends on the final sampling variability metric of interest. For example, pointwise standard errors for the PCs can be calculated more quickly than the full, high dimensional bootstrap distribution of the PCs. Section 3.4 uses simulations to demonstrate coverage rates for confidence regions around the PCs. Section 3.5 applies fast bootstrap PCA to the EEG and MRI datasets.

3.1.1 A brief summary of PCA, SVD, the bootstrap, and their accompanying notation

In the remainder of this paper, we will use the notation $\mathbf{X}_{[i,k]}$ to denote the element in the i^{th} row and k^{th} column of the matrix $\mathbf{X}$. The notation $\mathbf{X}_{[,k]}$ denotes the k^{th} column of $\mathbf{X}$; $\mathbf{X}_{[k,]}$ denotes the k^{th} row of $\mathbf{X}$; $\mathbf{X}_{[,1:k]}$ denotes the first k columns of $\mathbf{X}$; and $\mathbf{X}_{[1:k,1:k]}$ denotes the block of matrix $\mathbf{X}$ defined by the

intersection of the first k columns and rows. The notation $\mathbf{v}_{[j]}$ denotes the j^{th} element of the vector $\mathbf{v}$, the notation $\mathbf{1}_k$ denotes the k-length vector of ones, and the notation $\mathbf{I}_k$ denotes the $k \times k$ identity matrix. We will also generally use the term "orthonormal matrix" to refer to rectangular matrices with orthonormal columns.

In order to create highly informative feature variables, PCA determines the set of orthonormal basis vectors such that the subjects' coordinates with respect to these new basis vectors are maximally variable (Jolliffe, 2005). These new basis vectors are called the sample principal components (PCs), and the subjects coordinates with respect to these basis vectors are called the sample scores.

Both the sample PCs and sample scores can be calculated via the singular value decomposition (SVD) of the sample data matrix. Let $\mathbf{Y}$ be a full rank, $p \times n$ data matrix, containing p measurements from n subjects. Suppose that the rows of $\mathbf{Y}$ have been centered, so that each of the p dimensions of $\mathbf{Y}$ has mean zero. The singular value decomposition of $\mathbf{Y}$ can be denoted as $\mathbf{VDU}'$, where $\mathbf{V}$ is the $p \times n$ matrix containing the orthonormal left singular vectors of $\mathbf{Y}$, $\mathbf{U}$ is the $n \times n$ matrix containing the right singular vectors of $\mathbf{Y}$, and $\mathbf{D}$ is a $n \times n$ diagonal matrix whose diagonal elements contain the ordered singular values of $\mathbf{Y}$. The principal component vectors are equal to the ordered columns of $\mathbf{V}$, and the sample scores are equal to the $n \times n$ matrix $\mathbf{DU}'$. The diagonal elements of $(1/(n-1))\,\mathbf{D}^2$ contain the sample variances for each score variable, also known as the variances explained by each PC. Approximations of $\mathbf{Y}$ using only the first K principal components can be constructed as $\hat{\mathbf{Y}} := \sum_{k=1}^{K} \mathbf{V}_{[,k]}(\mathbf{DU}')_{[k,]}$. Existing methods for fast, exact, and scalable calculation of the SVD in high dimensional samples are discussed in the supplemental materials.

The sampling variability of PCA can be estimated using a bootstrap procedure. The first step of this procedure is to construct a bootstrap sample, by drawing n observations, with replacement, from the original demeaned sample. PCA is reapplied to the bootstrap sample, and the results are stored. This process is repeated B times, until B sets of PCA results have been calculated from B bootstrap samples. We index the bootstrap samples by the superscript notation b, so that $\mathbf{Y}^b$ denotes the b^{th} bootstrap sample. Variability of the PCA results across bootstrap samples is then used to approximate the variability of PCA results across different samples from the population. Unfortunately, recalculating the SVD for all B bootstrap samples has a computation complexity of order $O(Bpn^2)$, which can make the procedure computationally infeasible when p is very large.

3.1.2 Fast bootstrap PCA – resampling is a low dimensional transformation

It's important to note that the interpretation of principal components (PCs) depends on the coordinate vectors on which the sample is measured. Given the sample coordinate vectors, the PC matrix represents linear transformation that aligns the coordinate vectors with the directions along which sample points are most variable. When the number of coordinate vectors (p) exceeds the number of observations (n), this transformation involves first reducing the coordinate vectors to a parsimonious, orthonormal basis of n vectors[1] whose span still includes the sample data points, and then applying the unitary transformation

[1]Note, if the data has been centered, then $n-1$ basis vectors are sufficient. For brevity of notation though, we will generally refer to the subspace under either scenario, centered or uncentered, as n-dimensional.

that aligns this basis with the directions of maximum sample variance. The first step, of finding a parsimonious basis, is more computationally demanding than the alignment step. However, if the number of coordinate vectors is equal to the number of data points, then the transformation obtained from PCA consists of only an alignment.

The key to improving computational efficiency of PCA in bootstrap samples is to realize that all resampled observations are contained in the same low dimensional subspace as the original sample. Because the span of the principal components $\mathbf{V}$ includes all observations in the original sample, the span of $\mathbf{V}$ also includes all observations in any bootstrap sample. Thus, in each bootstrap sample, $\mathbf{Y}^b$, we can skip the computationally demanding dimension reduction step of the PCA by first representing $\mathbf{Y}^b$ in terms of the parsimonious, orthonormal basis $\mathbf{V}$. Viewing the bootstrap procedure as a loop operation over several bootstrap samples, we see that the low dimensional subspace on which all sample points lie is loop invariant.

To translate this intuition into the calculation of the SVD for bootstrap samples, we first note that $\mathbf{Y}^b$ can be represented as $\mathbf{YP}^b$, where $\mathbf{P}^b_{[i,j]} = 1$ if $\mathbf{Y}^b_{[,j]} = \mathbf{Y}_{[,i]}$ and zero otherwise. In each bootstrap sample, we then calculate its SVD, denoted by $\mathbf{V}^b\mathbf{D}^b\mathbf{U}^{b'}$, via the following steps

$$
\begin{array}{lll}
\mathbf{Y}^b & = \mathbf{YP}^b & \text{where } \mathbf{P}^b \text{ represents a resampling operation} \\
& = \mathbf{VDU}'\mathbf{P}^b & \text{where } \mathbf{DU}'\mathbf{P}^b \text{ is the matrix of resampled scores} \\
& = \mathbf{V}(\mathbf{A}^b\mathbf{S}^b\mathbf{R}^{b'}) & \text{where } \mathbf{A}^b\mathbf{S}^b\mathbf{R}^{b'} := svd(\mathbf{DU}'\mathbf{P}^b) \\
& = (\mathbf{VA}^b)\mathbf{S}^b(\mathbf{R}^b)' & \text{where } (\mathbf{VA}^b) \text{ and } (\mathbf{R}^b) \text{ are orthonormal, and } \mathbf{S}^b \text{ is diagonal} \\
& = svd(\mathbf{Y}^b) &
\end{array}
$$

Rather than directly decomposing the p-dimensional bootstrap sample $\mathbf{Y}^b$, we reduce the problem to a decomposition of the n-dimensional resampled

scores, $svd(\mathbf{DUP}^b) =: \mathbf{A}^b\mathbf{S}^b\mathbf{R}^{b'}$. Because $\mathbf{V}$ and $\mathbf{A}^b$ are both orthonormal, their product $\mathbf{VA}^b$ is orthonormal as well. Since $\mathbf{S}$ is diagonal and $\mathbf{R}^b$ is orthonormal, $(\mathbf{VA}^b)\mathbf{S}^b(\mathbf{R}^b)$ is equal to the SVD of $\mathbf{Y}^b$. The singular values, and right and left singular vectors of the $\mathbf{Y}^b$ can then be written respectively as $\mathbf{D}^b = \mathbf{S}^b$, $\mathbf{U}^b = \mathbf{R}^b$, and $\mathbf{V}^b = \mathbf{VA}^b$. If only the first K principal components are of interest, then it is sufficient to calculate and store $\mathbf{A}^b$, $\mathbf{U}^b$, and $\mathbf{D}^b$ as the matrices containing only the first K singular vectors and values of $\mathbf{DU}'\mathbf{P}^b$. Full details of our proposed methods for bootstrap PCA are discussed in section 3.3.

Daudin et al. (1988) applied an equivalent result to eigen-decompositions of bootstrap covariance matrices in the $p < n$ setting, but this result has not been widely used, nor has it been generalized to the $p >> n$ setting. Daudin et al. (1988) suggested that, rather than decomposing the $p \times p$ covariance matrix, a more computationally efficient approximation is to decompose the covariance matrix of the k leading resampled score variables. The eigenvectors of this $k \times k$ covariance matrix can then be projected onto the p-dimensional space to approximate the eigenvectors of the full $p \times p$ covariance matrix. In the $p >> n$ setting, however, if k is set equal to n, then the approximation becomes exact. Note also that in the $p >> n$ setting, it is the projection onto the p-dimensional space that is most computationally demanding step (computational complexity $O(KBpn)$), rather than the n-dimensional decompositions (computational complexity $O(KBn^2)$).

To gain intuition for why that the columns of $\mathbf{VA}^b$ are the principal components of $\mathbf{Y}^b$, note that the resampled scores, $\mathbf{DU}'\mathbf{P}^b$, are equivalent to the resampled data, $\mathbf{Y}^b$, expressed in terms in terms of the coordinate vectors $\mathbf{V}$. This implies that the principal components of the resampled scores, $\mathbf{A}^b$, give the

transformation required to align the coordinate vectors of the scores, $\mathbf{V}$, with the directions along which the resampled scores are most variable. Applying this transformation yields $\mathbf{V}\mathbf{A}^b$ – the bootstrap principal components in terms of the sample's original, native coordinate vectors.

Random orthogonal rotations comprise the only possible way that the fitted PCs can vary across bootstrap samples. Because of this, the bootstrap procedure will not be able to directly estimate PC sampling variability in directions orthogonal to the observed sample, not unlike how a bootstrap mean estimate must be a weighted combination of the observed data points. However, when the inherent dimension of the population is small, the sampling variability of the PCs will generally be dominated by variability in a handful of directions, and these directions will generally be well represented by the span of the bootstrap PCs. Variability in directions not captured by the bootstrap procedure will tend to be of a much smaller magnitude.

The rotational variability of the bootstrap PCs is directly represented by the $\mathbf{A}^b$ matrices. More specifically, information about random rotations within the K leading PCs is captured by the $\mathbf{A}^b_{[1:K,1:K]}$ block matrices, which show how much each of the K leading bootstrap PCs weight on each of original K leading components. When the majority of bootstrap PC variability is due to rotations within the K leading PCs, the $\mathbf{A}^b_{[1:K,1:K]}$ matrices provide a parsimonious description of this dominant form of variability.

Decomposing $\mathbf{V}^b$ into an alignment operation, $\mathbf{A}^b$, applied to the original sample components, $\mathbf{V}$, can drastically reduce the storage and memory requirements required for the bootstrap procedure, making it much more amenable to parallelization. Using this method, we're able to store all the information

about the variability of $\mathbf{V}^b$ only by storing the $\mathbf{A}^b$ matrices, which can later be projected onto the high dimensional space. Calculating the $\mathbf{A}^b$ matrices only requires the low dimensional matrices $\mathbf{DU}'$ and $\mathbf{P}^b$, and does not require either operations on the $p \times n$ matrix $\mathbf{Y}^b$, or access to the potentially large data files storing $\mathbf{Y}$. In the context of parallelizing the bootstrap procedure, this allows for minimal memory, storage, and data access requirements for each computing node.

Furthermore, in many cases, it is not even necessary to calculate and store the p-dimensional components, $\mathbf{V}^b_{[,1:K]}$. Instead we can calculate summary statistics for the bootstrap distribution of the low dimensional matrices $\mathbf{A}^b$, and translate only the summary statistics to the high dimensional space. For example, we can quickly calculate bootstrap standard errors for $\mathbf{V}_{[,1:K]}$ by first calculating the bootstrap moments of $\mathbf{A}^b$, and projecting these moments back onto the p-dimensional space (see section 3.3.2). Joint confidence regions for the PCs can also be constructed solely based on the bootstrap distribution of $\mathbf{A}^b$ (see section 3.3.3). Similar complexity reductions are available when calculating bootstrap distribution of linear functions of the components, such as the the arithmetic mean of the k^{th} PC (i.e. $(1/p)\mathbf{1}_p'\mathbf{V}^b_{[,k]}$). For any bootstrap statistic of the form $\mathbf{q}'\mathbf{V}^b_{[,k]} = (\mathbf{q}'\mathbf{V})\mathbf{A}^b_{[,k]}$, where $\mathbf{q}$ is a p-length vector, the n-length vector $\mathbf{q}'\mathbf{V}$ can be pre-calculated, and the complexity of the bootstrap procedure will be limited only by n.

3.2 Motivating data

In this section we apply standard PCA to a dataset of sleep EEG recordings (p=900), and to a dataset of preprocessed brain MRIs (p=2,979,666). A bootstrap procedure is later applied in section 3.5, to estimate sampling variability for the fitted PCs.

There has been demonstrated interest in the population PCs corresponding to both datasets (Di et al., 2009; Crainiceanu et al., 2011; Zipunnikov et al., 2011a,b). For our purposes, the functional EEG data form an especially useful didactic example, as the sample PCs are also functional, and easily visualizable. We include the MRI dataset primarily to demonstrate computational feasibility of the bootstrap procedure when dimension (p) is large.

3.2.1 Sleep EEG

The Sleep Heart Health Study (SHHS) is a multi-center prospective cohort study, designed to analyze the relationships between sleep-disorder breathing, sleep metrics, and cardiovascular disease (Quan et al., 1997). Along with many other health and sleep measurements, EEG recordings were taken for each patient, for an entire night's sleep. An EEG uses electrodes placed on the scalp to monitor neural activation in the brain, and is commonly used to describe the stages of sleep. Our goal in this application is to estimate the primary patterns in EEG signal that differentiate among healthy subjects, and to quantify uncertainty in these estimated patterns due to sampling variability.

To reflect this goal, we selected a subsample of 392 healthy, comparable

controls from the SHHS ($n = 392$). Our sample contained only female participants between ages 40 and 60, with no sleep disordered breathing, no history of smoking, and high quality EEG recordings for at least 7.5 hours of sleep. In order to more easily register EEG recordings across subjects, only the first 7.5 hours of EEG data from each subject were used. Although the EEG recordings consist of measurements from two electrodes, we focus for simplicity only on measurements from one of these electrodes (from the left side of the top of the scalp).

To process the raw EEG data, each subject's measurements were divided into thirty second windows, and the proportion of the signal in each window attributable to low frequency wavelengths (0.8-4.0 Hz) was recorded. This proportion is known as normalized δ power (NP_δ), and is particularly relevant to deep stage sleep (NREM Stage 3). The preprocessing procedure used here to transform the raw EEG data into NP_δ is the same as the procedure used by Crainiceanu et al. (2009). A lowess smoother was then applied to each subject's NP_δ function, as a simple means of incorporating the assumption that the underlying NP_δ process is a smooth function. This preprocessing procedure resulted in 7.5 hours $\times$ (60 minutes / hour) $\times$ (2 thirty second windows / minute) = 900 measurements of NP_δ per subject ($p = 900$).

The left panel of Figure 3.1 shows examples of NP_δ functions for five subjects, as well as the mean NP_δ function across all subjects, denoted by $\boldsymbol{\mu}$. The first five principal components of the NP_δ data are shown in the right panel of Figure 3.1. The first PC (PC1) appears to be a mean shift, indicating that the primary way in which subjects differ is in their overall NP_δ over the course of the night. The remaining four PCs (PC2, PC3, PC4, and PC5) roughly

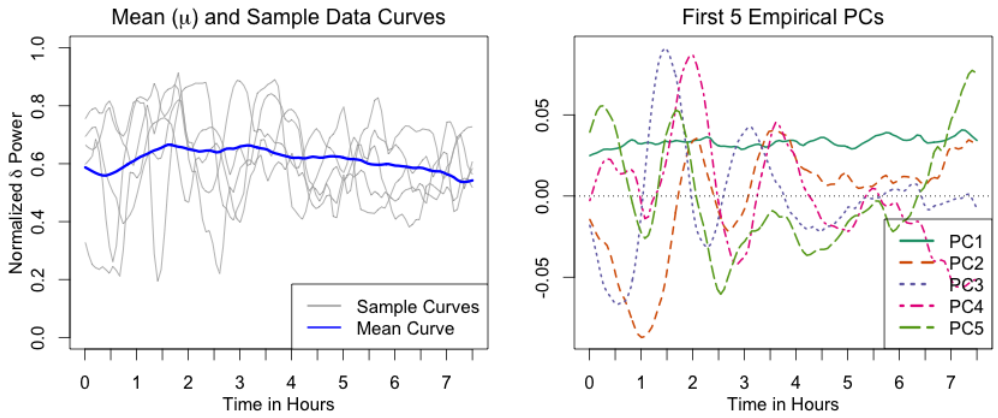

Figure 3.1: Summary of EEG dataset - The left panel shows examples of normalized δ power (NP_δ) over the course of the night for five subjects, as well as the mean NP_δ function across all subjects ($\boldsymbol{\mu}$). The right panel shows the first five PCs of the dataset.

correspond to different types of oscillatory patterns in the early hours of sleep. These components are fairly similar to the results found by (Di et al., 2009), who analyze a different subset of the data, and employ a smooth multilevel functional PCA approach to estimate eigenfunctions that differentiate subjects from one another.

Collectively, the first five PCs explain approximately 55% of the variation, and the first ten PCs explain approximately 76% of the variation (see scree plot in supplemental materials). These estimates for the variance explained by each component are much lower than the estimates from Di et al. (2009). The difference is most likely due differences in how the MFPCA method employed by Di et al. (2009) incorporates the assumption of underlying smoothness in NP_δ.

3.2.2 Brain magnetic resonance images

We also consider a sample data processed using voxel based morphometry (VBM) (Ashburner and Friston, 2000), a technique that is frequently used to study differences in the size of brain regions across subjects, or within a single subject over time. Our data came from an epidemiological study of former organolead manufacturing workers (Stewart et al., 2006; Schwartz et al., 2007, 2010; Bobb et al., 2014). We focused on the baseline MRIs from the 352 subjects for which both baseline and followup MRIs were recorded.

VBM images were constructed based on brain MRIs. The original MRIs were stored as 3-dimensional arrays, with each array element corresponding to tissue intensity in a voxel, or volumetric pixel, of the brain. Creating VBM images typically begins by registering each subject's brain MRI to a common template image, using a non-linear warping. The number of voxels mapped to each voxel of the template image during the registration process is recorded. This information is used to create subject-specific images on the template space, where each voxel's intensity represents the size of that voxel in the subject's original MRI. The VBM images used here were processed using a generalization of the regional analysis of volumes examined in normalized space (RAVENS) algorithm (Goldszal et al., 1998; Davatzikos et al., 2001), and are the same as the baseline visit images used in (Zipunnikov et al., 2011b,a).

To create a single $p \times n$ data matrix, each subject's VBM image was vectorized, omitting the background voxels that did not correspond to brain tissue. The vector for each subject contained 2,979,666 measurements (p=2,979,666).

Because the resulting data matrix was 3.5 Gb, it is difficult to store the entire data matrix in working memory, and block matrix algebra is required to calculate the sample PCs (see supplemental materials).

A central slice from each of the first three PCs is shown in the first row of Figure 3.3. PC1 appears to roughly correspond with grey matter, indicating that the primary way in which subjects regions tend to differ is in their overall grey matter volume. Together, the first 30 PCs explain approximately 53.3% of the total sample variation (see scree plot in supplemental materials).

In the remainder of this paper, we refer to this dataset primarily as to demonstrate the computational feasibility of bootstrap PCA in especially high dimensions. Additional interpretation of the sample PCs is given in (Zipunnikov et al., 2011b,a).

3.3 Full description of the bootstrap PCA algorithm

In this section we outline calculation methods for bootstrap standard errors, bootstrap confidence regions, and for the full bootstrap distribution of the principal components (PCs). The overall computational complexity of the procedure depends on the bootstrap metric of interest, but the initial steps of all our proposed methods are the same.

Building on the notation of sections 3.1.1 and 3.1.2, let K be the number of principal components that are of interest, which typically will be less than $n-1$. For simplicity of presentation, we assume that each dimension of the bootstrap sample $\mathbf{Y}^b$ has mean zero. Manually recentering $\mathbf{Y}^b$ however will not add any high dimensional complexity to the procedure, as this is equivalent

to recentering the $n \times n$ matrix of resampled scores $\mathbf{DU}'\mathbf{P}^b$ (see supplemental materials).

For each bootstrap sample, we begin by calculating the leading K singular vectors and singular values of the resampled scores $\mathbf{DU}'\mathbf{P}^b$. As noted in section 3.1.2, the leading left and right singular vectors of $\mathbf{DU}'\mathbf{P}^b$ are stored as solutions for the $n \times K$ matrices $\mathbf{A}^b$ and $\mathbf{U}^b$ respectively. The leading singular values of $\mathbf{DU}'\mathbf{P}^b$ are the solutions for the diagonals of the $K \times K$ matrix $\mathbf{D}^b$. In the typical case where K is less than or equal to the rank of $\mathbf{DU}'\mathbf{P}^b$, the first K singular values of $\mathbf{DU}'\mathbf{P}^b$ are positive and unique, and the solutions for the columns of $\mathbf{A}^b$ and $\mathbf{U}^b$ are unique up to sign changes. Arbitrary sign changes in the columns of $\mathbf{A}^b$ will ultimately result in arbitrary sign changes in the bootstrap PCs. Adjusting for these arbitrary changes is discussed in section 3.3.1.

We find in approximately 4% of bootstrap samples from the MRI dataset, that although a solution to the SVD of $\mathbf{DU}'\mathbf{P}^b$ exists, the SVD function fails to converge. We handle these cases by randomly preconditioning the matrix $\mathbf{DU}'\mathbf{P}^b$, reapplying the SVD function, and appropriately adjusting the results to find the SVD of the original matrix. The full details of this procedure are described in the supplement materials.

These baseline steps require a computational complexity of order $O(KBn^2)$. They are sufficient for calculating the leading K bootstrap scores and the variance explained by the leading K bootstrap PCs.[2]

When moving on to describe the bootstrap distribution of the PCs, we have

[2]The bootstrap score matrix is equal to $\mathbf{D}^b\mathbf{U}^{b'}$, and the variances explained by each bootstrap PC are equal to the diagonals of $(1/(n-1))(\mathbf{D}^b)^2$. These variances explained can also be expressed as a proportions of the total variance of the bootstrap sample, which can be calculated as $trace(Var(\mathbf{Y}^b)) = (1/(n-1))||\mathbf{DU}'\mathbf{P}^b||^2 = (1/(n-1))\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{DU}'\mathbf{P}^b)^2_{[i,j]}$.

several options, each requiring a different level of computational complexity:

- **Standard errors for the PCs** can be calculated based on the bootstrap mean and variance of the columns of $\mathbf{A}^b$ (see section 3.3.2). These standard errors can be used to create pointwise confidence intervals (see section 3.3.3). This option requires additional computational complexity of order $O(Kpn^2 + KBn^2)$.

- **Joint confidence regions for the PCs** and for the principal subspace can be constructed using the methods in section 3.3.3. This option requires no additional computational complexity on the high dimensional scale.

- **The full bootstrap distribution of PCs** can be calculated by projecting the principal components of the bootstrap scores onto the p-dimensional space (i.e. $\mathbf{V}^b_{[,1:K]} = \mathbf{V}\mathbf{A}^b$). The bootstrap PC vectors ($\mathbf{V}^b_{[,1:K]}$) can then be used to create pointwise percentile intervals for the PCs (see section 3.3.3). If p is sufficiently large such that the matrix $\mathbf{V}$ cannot be held in working memory, block matrix algebra can be used to break down the calculation of $\mathbf{V}\mathbf{A}^b$ into a series low memory operations (see supplemental materials). Calculation of all bootstrap PCs requires additional computational complexity of order $O(KBpn)$. If K is set equal to $n-1$, then the computational complexity of this method is roughly equivalent to that the standard methods ($O(Bpn^2)$). The total computation time, however, will still be approximately half the time of standard methods, as the matrices $\mathbf{Y}^{b'}\mathbf{Y}^b$ need not be calculated (see supplemental materials).

3.3.1 Adjusting for axis reflections of the principal components

Because the singular vectors of $\mathbf{Y}^b$ are not unique up to sign, arbitrary sign changes, also known as reflections across the origin, will induce variability in both the sampling and bootstrap distributions of the principal components ($\mathbf{V}^b$). These reflections, however, do not affect the interpretation of the PCs, and so their induced variability will cause us to overestimate sampling variability of the patterns decomposed by PCA (Efron and Tibshirani, 1993, see section 7.2; Mehlman et al., 1995; Jackson, 1995; Milan and Whittaker, 1995). For example, arbitrary sign changes can cause the confidence interval for any element of any principal component to include zero, even if the absolute value of that element is nearly constant and nonzero across all bootstrap samples.

To isolate only the variation that affects the interpretation of the PCs, we adjust the sign of the columns of $\mathbf{V}^b$ so that the dot products $\mathbf{V}^{b'}_{[,k]}\mathbf{V}_{[,k]}$ are positive for $k = 1, 2, \ldots, K$. Note that because $\mathbf{V}^b = \mathbf{V}\mathbf{A}^b$, sign changes in the columns of $\mathbf{V}^b$ are equivalent to sign changes in the columns of $\mathbf{A}^b$. For the same reason, sign adjustments for the columns of $\mathbf{V}^b$ are equivalent to sign adjustments for the columns of $\mathbf{A}^b$, which can be simpler to compute. Here, the dot products $\mathbf{V}^{b'}_{[,k]}\mathbf{V}_{[,k]}$ for $k = 1, 2, \ldots, K$ actually do not require any additional calculations, as they can be found on the diagonal elements of $\mathbf{V}'\mathbf{V}^b = \mathbf{V}'\mathbf{V}\mathbf{A}^b = \mathbf{A}^b$. Independent of our work, this calculation simplification is also noted by Daudin et al. (1988). Whenever $\mathbf{A}^b_{[k,k]}$ is negative, we declare that an arbitrary sign change has occurred, and adjust by multiplying $\mathbf{A}^b_{[,k]}$ and $\mathbf{U}^b_{[,k]}$ by -1. The resulting PCs and scores are still valid solutions to the PCA

algorithm.

Since $\mathbf{V}^b_{[,k]}$ and $\mathbf{V}_{[,k]}$ each have norm equal to one, their dot product is equal to the cosine of the angle between them. As a result, using the dot product $\mathbf{V}^{b'}_{[,k]}\mathbf{V}_{[,k]}$ to adjust for sign will ensure that the angle between $\mathbf{V}^b_{[,k]}$ and $\mathbf{V}_{[,k]}$ is between $-\pi/2$ and $\pi/2$. This range of angles is exactly the range that affects our interpretation of the bootstrapped PCs. Using these dot products for sign adjustment is also equivalent to choosing the sign of $\mathbf{V}^b_{[,k]}$ that minimizes the Frobenius distance $||\mathbf{V}^b_{[,k]} - \mathbf{V}_{[,k]}||$, a method that has been previously suggested (Lambert et al., 1991; Milan and Whittaker, 1995).

It has also been suggested that the sign of each PC should be switched based on the correlation between the columns of $\mathbf{V}^b$ and the columns of $\mathbf{V}$, rather than the dot products $\mathbf{V}^{b'}_{[,k]}\mathbf{V}_{[,k]}$ (Jackson, 1995; Babamoradi et al., 2012).[3] We advocate against this correlation method, in favor of the cross product method. Of course, the two methods are very similar, as the correlation method is equivalent to applying a cross product operation after first centering and scaling the two vectors. Pre-scaling has no practical effect, as only the sign of the correlation is retained. However, pre-centering removes information that is potentially relevant to the sign switch decision. For example, consider the case where $\mathbf{V}_{[,k]}$ is proportional to a sine wave, shifted up by 2, and scaled appropriately to have norm 1. Furthermore, let $\mathbf{V}^b_{[,k]}$ be proportional to the same sine wave, shifted down by 2, and similarly scaled to have norm 1. Note that $\mathbf{V}_{[,k]}$ has all positive elements, and $\mathbf{V}^b_{[,k]}$ has all negative elements. These two vectors will be positively correlated, but have a negative crossproduct. The correlation rule will

[3]Here, the correlation operation is taken across the p elements of the vector, without the operation's common statistical interpretation that each vector element is a new observation of a random variable.

not result in a sign change, which can yield a bimodal bootstrap PC distribution with PCs clustered on either side of the zero line. Alternatively, the cross product rule will result in a sign change, making a bimodal bootstrap distribution less likely. In the supplemental materials, we illustrate such cases in more detail, and further argue for the use of the cross product over the correlation.

3.3.2 Bootstrap moments of the principal components

Traditional calculation of the mean and variance of $\mathbf{V}^b_{[,k]}$ requires first calculating the bootstrap distribution of $\mathbf{V}^b_{[,k]}$, and then taking means and variances over all B bootstrap samples. However, using our characterization of $\mathbf{V}^b$ as $\mathbf{V}\mathbf{A}^b$, and properties of expectations, the same result can achieved without calculating or storing $\mathbf{V}^b_{[,k]}$.

Specifically, the bootstrap mean $E(\mathbf{V}^b_{[,k]})$ can be found via $E(\mathbf{V}\mathbf{A}^b_{[,k]}) = \mathbf{V}E(\mathbf{A}^b_{[,k]})$, where the operation E is the expectation with respect to the bootstrap distribution. The bootstrap variance of $\mathbf{V}^b_{[i,k]}$ can be found via

$$Var(\mathbf{V}^b_{[i,k]}) = Cov(\mathbf{V}^b_{[,k]})_{[i,i]} = Cov(\mathbf{V}\mathbf{A}^b_{[,k]})_{[i,i]} = [\mathbf{V}Cov(\mathbf{A}^b_{[,k]})\mathbf{V}']_{[i,i]} = (\mathbf{V}_{[i,]})'Cov(\mathbf{A}^b_{[,k]})(\mathbf{V}_{[i,]})$$

Where Var and Cov are variance operators with respect to the bootstrap distribution. The total computational complexity of finding $Cov(\mathbf{A}^b_{[,k]})$ and then $Var(\mathbf{V}^b_{[i,k]})$ for each combination of $i = 1, 2, \ldots, p$ and $k = 1, \ldots, K$ is only $O(Kpn^2 + KBn^2))$.[4]

[4]In practice, we calculate the diagonals of $\mathbf{V}Cov(\mathbf{A}^b_{[,k]})\mathbf{V}'$ by the row sums of $(\mathbf{V}Cov(\mathbf{A}^b_{[,k]})) \circ (\mathbf{V})$, where $\circ$ denotes element-wise multiplication as opposed to traditional matrix multiplication.

This improvement in computation speed comes from pre-collapsing the complexity induced by having a large number of bootstrap samples before transforming to the high dimensional space. This allows us to separate calculations of order B from calculations of order p. Similar speed improvements are attainable whenever summary statistics or parametric models for the bootstrap distribution of $\mathbf{A}^b$ can be translated into summary statistics or parametric models for the high dimensional components $\mathbf{V}^b$.

3.3.3 Construction of confidence regions

Several types of confidence regions can be constructed based on the bootstrap distribution the PCs. In this section, we specifically discuss (1) pointwise confidence intervals (CIs) for the PCs, based on either the bootstrap moments or bootstrap percentiles; (2) confidence regions (CRs) for the individual PCs; and (3) CRs for the principal subspace. Only the pointwise percentile intervals require calculation of the full bootstrap distribution of the high dimensional PCs. All other CRs can be calculated solely based on the bootstrap distribution of the low dimensional $\mathbf{A}^b$ matrices.

Pointwise confidence intervals for the principal components

The simplest pointwise confidence interval for the principal components is the moment-based, or Wald confidence interval. For the i^{th} element of the k^{th} PC, the moment-based CI is defined as $E(\mathbf{V}^b_{[i,k]}) \pm \sigma(\mathbf{V}^b_{[i,k]}) z_{(1-\alpha/2)}$, where α is the desired alpha level, $z_{(1-\alpha/2)}$ is the $100(1-\alpha/2)^{th}$ percentile of the standard normal distribution, and the E and σ functions capturing the mean and standard deviation of $\mathbf{V}^b_{[i,k]}$ across bootstrap samples. Both $E(\mathbf{V}^b_{[i,k]})$ and $\sigma(\mathbf{V}^b_{[i,k]})$ can be

attained without calculating or storing the full bootstrap distribution of $\mathbf{V}^b_{[i,k]}$ (see section 3.3.2).

Another common pointwise interval for $\mathbf{V}^b_{[i,k]}$ is the bootstrap percentile CI, defined as $(q(\mathbf{V}^b_{[i,k]}, \alpha/2), q(\mathbf{V}^b_{[i,k]}, 1-\alpha/2))$, where $q(\mathbf{V}^b_{[i,k]}, \alpha)$ denotes the $100\alpha^{th}$ percentile of the bootstrap distribution of $\mathbf{V}^b_{[i,k]}$. Unlike the moment-based CI, the percentile CI does require calculation of the full bootstrap distribution of $\mathbf{V}^b_{[i,k]}$.

Estimating the percentile interval tends to require more bootstrap samples (e.g. B=1000-2000) than estimating the moment-based interval (e.g. B=50-200), as the quantile function is more affect affected by the tails of the bootstrap distribution than the moments are (Efron and Tibshirani, 1993). Interpretation of both these pointwise CIs is discussed further in section 3.5.

Our methods can also be used to quickly calculate bias corrected and accelerated (BC_a) CIs (Efron, 1987), as others have suggested (Timmerman et al., 2007;Salibián-Barrera et al., 2006).

Confidence regions for the principal components

Each principal component can be represented as a point in p-dimensional space. More specifically, because of the norm 1 requirement for the PCs, the parameter space for the principal components is restricted to the p-dimensional unit hypersphere, $S_p = \{\mathbf{x} \in R^p : \mathbf{x}'\mathbf{x} = 1\}$. To create p-dimensional CRs for each PC, Beran and Srivastava (1985) suggest so-called confidence cones on the unit hypersphere, of the form

$$\{\mathbf{x} \in S_p : |\mathbf{x}'\mathbf{V}_{[,k]}| \geq q(|\mathbf{V}^{b'}_{[,k]}\mathbf{V}_{[,k]}|, \alpha) = q(|\mathbf{A}^b_{[k,k]}|, \alpha)\}$$

Here, $q(a^b, \alpha)$ is the quantile function denoting the $100\alpha^{th}$ bootstrap percentile of the statistic a^b. As noted in section 3.3.1, the calculation of $\mathbf{V}^{b'}_{[,k]}\mathbf{V}_{[,k]}$ can be simplified to $\mathbf{V}^{b'}_{[,k]}\mathbf{V}_{[,k]} = \mathbf{A}^{b}_{[k,k]}$. Geometrically, the dot product condition of this CR is equivalent to a condition on the angle between $\mathbf{x}$ and $\mathbf{V}_{[,k]}$. Note that this CR automatically incorporates the sign adjustments described in section 3.3.1. Beran and Srivastava (1985) provide a theoretical proof for the coverage of CRs constructed in this way.

It is also possible to create joint confidence bands (jCBs) for the PCs according the method outlined by Crainiceanu et al. (2012). However, such bands will also contain vectors that do not have norm 1, and may even exceed 1 in absolute value for a specific dimension. As a result, many vectors contained within the jCBs will not be valid principal components, which complicates interpretation of the jCBs.

Confidence regions for the principal subspace

To characterize the variability of the subspace spanned by the first K PCs, also known as the principal subspace, it is not sufficient to simply combine the individual CRs for each PC. This is because the sampling variability of the individual fitted PCs is influenced by random rotations of the fitted PC matrix $\mathbf{V}^{b}_{[,1:K]}$, while the sampling variability of the subspace is not. Similarly, most models whose fit depends on the leading PCs are unaffected by random rotations.

To characterize the sampling variability of the principal subspace, we first note that any bootstrap principal subspace can be defined by the $p \times K$ matrix with columns equal to the leading K PCs. Any such matrix must be contained

within the set of all of $p \times K$ orthonormal matrices. This set can be written as the Stiefel manifold $M_K(R^p) := \{\mathbf{X} \in F^{p\times K} : \mathbf{X}'\mathbf{X} = \mathbf{I}_K\}$, where $F^{p\times K}$ is the set of all $p \times K$ matrices. To create CRs for the principal subspace, we can use the following generalization of CRs for the individual PCs

$$\{\mathbf{X} \in M_K(R^p) : \ ||\mathbf{X}'\mathbf{V}_{[,1:K]}|| \geq q(||\mathbf{V}^{b'}_{[,1:K]}\mathbf{V}_{[,1:K]}||, \alpha) = q(||\mathbf{A}^{b}_{[1:K,1:K]}||, \alpha)\}$$

Here, the norm operation refers to the Frobenius norm. Beran and Srivastava (1985) suggest CRs of this form to characterize variability of a set of sample covariance matrix eigenvectors whose corresponding population eigenvalues are all equal. However, the CR construction method can also be applied in the context of estimating the principal subspace. As with CRs for the individual PCs, we can make the simplification that $\mathbf{V}^{b'}_{[,1:K]}\mathbf{V}_{[,1:K]} = \mathbf{A}^{b}_{[1:K,1:K]}$. Note that such CRs automatically adjust for random rotations of the first K principal components – if $\mathbf{R}$ is a $K \times K$ orthonormal transformation matrix, then $||(\mathbf{XR})'\mathbf{V}|| = ||\mathbf{X}'\mathbf{V}||$.

3.3.4 Maintaining informative rotational variability

When several of the leading eigenvalues of the population covariance matrix are close, the fitted PCs in any sample may be a mixtures the leading population PCs. In these cases, the bootstrap PCs will often be approximate rotations of the leading sample PCs. Others have argued if the parameter of interest is the principal subspace or the model fit, then the bootstrap PCs should be adjusted to correct for rotational variability, as the principal subspace is unaffected by rotations among the leading PCs. Specifically, it has been suggested to use a Procrustean rotation to match the bootstrap PCs to the original sample PCs (Milan and Whittaker, 1995), and to then create pointwise confidence intervals

(CIs) based on the rotated PCs (Timmerman et al., 2007; Babamoradi et al., 2012).[5] We argue however that bootstrap rotational variability is informative of genuine sampling rotational variability, and that adjusting for rotations is not an appropriate way to represent sampling variability of the principal subspace, or the sampling variability of model fit. This is because pointwise CIs are not designed to estimate the sampling variability of the principal subspace. The pointwise CIs generated from rotated bootstrap PCs also do not capture the sampling variability of standard PCs, as the rotated PCs are not valid solutions to the PCA algorithm.

Rather than rotating towards the sample, it has also been proposed to rotate both the sample and bootstrap PCs towards a $p \times K$ target matrix $\mathbf{T}$, which is pre-specified before collecting the initial sample $\mathbf{Y}$ (Raykov and Little, 1999; Timmerman et al., 2007).[6] The target matrix $\mathbf{T}$ may be based on scientific knowledge, or previous research. Such an approach can also be used to test null hypotheses about the principal subspace by rotating $\mathbf{V}^b_{[,1:K]}$ toward a null PC matrix $\mathbf{V}_0$ (Raykov and Little, 1999), and checking if elements of $\mathbf{V}_0$ are contained in the resulting CRs.

Our opinion is that if investigators are interested in the sampling variability of the output from a model that uses PCA, then it is the model output, and not the principal components, for which CRs should be calculated. If the sampling variability of the subspace is of interest, than CRs specifically designed for the

[5]One interpretation of CIs constructed from rotation adjusted bootstrap PCs is that if the population PC matrix is rotated towards the each sample from the population, then average pointwise coverage of rotation adjusted CIs should be approximately $100\alpha\%$

[6]The computational complexity of finding the appropriate rotation matrix in each bootstrap depends on the taking the SVD of the $K \times K$ matrix $\mathbf{V}^{b'}_{[,1:K]}\mathbf{T} = \mathbf{A}^{b'}_{[,1:K]}\mathbf{V}'\mathbf{T}$, where $\mathbf{V}'\mathbf{T}$ can be pre-calculated before the bootstrap procedure.

subspace should be used (see section 3.3.3), rather than adjusted CIs for the elements of the PCs. Rotating towards a pre-specified target matrix $\mathbf{T}$ can also be a useful approach, although it may be more interpretable to calculate the bootstrap distribution of the variance explained by the columns of $\mathbf{T}$,[7] rather than the bootstrap distribution of the fitted PCs after a rotation towards $\mathbf{T}$.

3.4 Coverage rate simulations

In this section we present simulated coverage rates for the CRs described in section 3.3.3. In order to make these simulations as realistic as possible, we simulated data using the empirical PC vectors of the EEG dataset as the true population basis vectors. As a baseline simulation scenario we set the sample size (n) equal to 392, and the true number of basis vectors in the population (denoted by K_0) equal to 5.

Measurement vectors for each subject were simulated according to the model $\mathbf{y}_i = \sum_{k=1}^{K_0} s_{ik}\mathbf{\Psi}_k + \boldsymbol{\epsilon}_i$, where $\mathbf{y}_i$ is a p-length vector of simulated measurements for the i^{th} subject; $\mathbf{\Psi}_k$ is the k^{th} true underlying basis vector, which is set equal to the k^{th} empirical PC of the EEG dataset; s_{ik} is a random draw from the empirical, univariate distribution of the scores for the k^{th} PC; and $\boldsymbol{\epsilon}_i$ is a vector of independent random normal noise variables, each with mean 0 and variance σ^2/p. Setting the variance of $\boldsymbol{\epsilon}_i$ equal to σ^2/p implies that the total variance attributable to the random noise will be approximately equal to σ^2, and will not depend on the number of measurements (p). The parameter σ^2 was set equal to

[7] In each bootstrap sample, the variance explained by the columns of $\mathbf{T}$ is equal to the variance of the resampled data after a projection onto the space spanned by $\mathbf{T}$. The projected data is equal to $\mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\mathbf{Y}^b = (\mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\mathbf{V})\mathbf{D}\mathbf{U}'\mathbf{P}^b$, where $\mathbf{T}'(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\mathbf{V}$ is an $n \times n$ matrix that can be precalculated before the bootstrap procedure.

the sum of the variances of the K_0+1 to n^{th} empirical score variables, implying that for each simulated sample, the first K_0 basis vectors $(\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_{K_0})$ were expected to explain approximately the same proportion of the variance that they explained in the empirical sample. For each simulated subject, $\mathbf{y}_i$, the K_0 score variables $s_{i1}, \ldots, s_{iK_0}$ were all drawn independently. Coverage was compared across 1000 simulated samples. For each simulated sample, the number of bootstrap samples created for estimation (B) was set to 1000.

As comparison simulation scenarios, we increased the number of measurements (p) to 5000 and to 20000, by interpolating the empirical EEG data and recalculating the principal components and scores. We also compared against simulated sample sizes (n) of 100 and 250. Because much of the variability in fitting principal components is determined by the spacing of eigenvalues in the population, we simulated separate scenarios where the empirical score distribution was scaled so that each basis vector explained half as much variance as the preceding basis vector. In other words, we scaled true population distribution of scores such that the vector of variances of the 5 score variables was proportional to the vector $(2^4, 2^3, 2^2, 2^1, 1)$. The total variance of the first 5 score variables was kept constant across all simulations. We refer to the modified eigenvalue spacing as the "parametric spacing" simulation scenario, and refer to the original eigenvalue spacing as the "empirical spacing" simulation scenario. Finally, we also simulated scenarios where the total variance due to the random noise (σ^2) was scaled up 50%, and where it was scaled down by 50%. Considering all combinations of eigenvalue spacing, random noise level, sample size, and number of measurements, we conducted $2 \times 3 \times 3 \times 3 = 54$ sets of simulations. Thus, our simulation study required the calculation of $54 \times 1000 \times 1000 = 54$ million

principal component decompositions, with the ranges for p and n mentioned above.

The total elapsed computation time for these 54 simulations was 28 hours. The simulations were run as a series of parallel jobs on an x86-based linux cluster, using a Sun Grid Engine for management of the job queue. As many as 200 jobs were allowed to run simultaneously. Each job required between approximately .5Gb and 2Gb maximum virtual memory, depending on the scenario being simulated.

3.4.1 Simulation results

The left of Figure 3.2 compares simulation results across different levels of residual variance, sample size, and eigenvalue spacing. In this 3×2 array of plots we fix p at 900, but results were similar for alternate values of p. For each simulation scenario, we calculated the median pointwise CI coverage across all 900 measurements. Both the moment-based and percentile intervals generally perform well, with all 54 simulation scenarios (including those not shown here) having median coverage rates between 92.4% and 98.1%. When the eigenvalues of the estimated PCs are well spaced (e.g. for PC1 in the empirical spacing scenario, or PCs 1-3 in the parametric spacing scenario), the coverage rates converge to 95% as the sample size increases. However, when the eigenvalues are not clearly differentiated, higher sample sizes can lead to slightly overly conservative CIs.

In the supplemental materials we further explore coverage by examining the full distribution of coverage rates across each of the p dimensions of the PCs, rather than summarizing by taking the median. We find that for both PC2

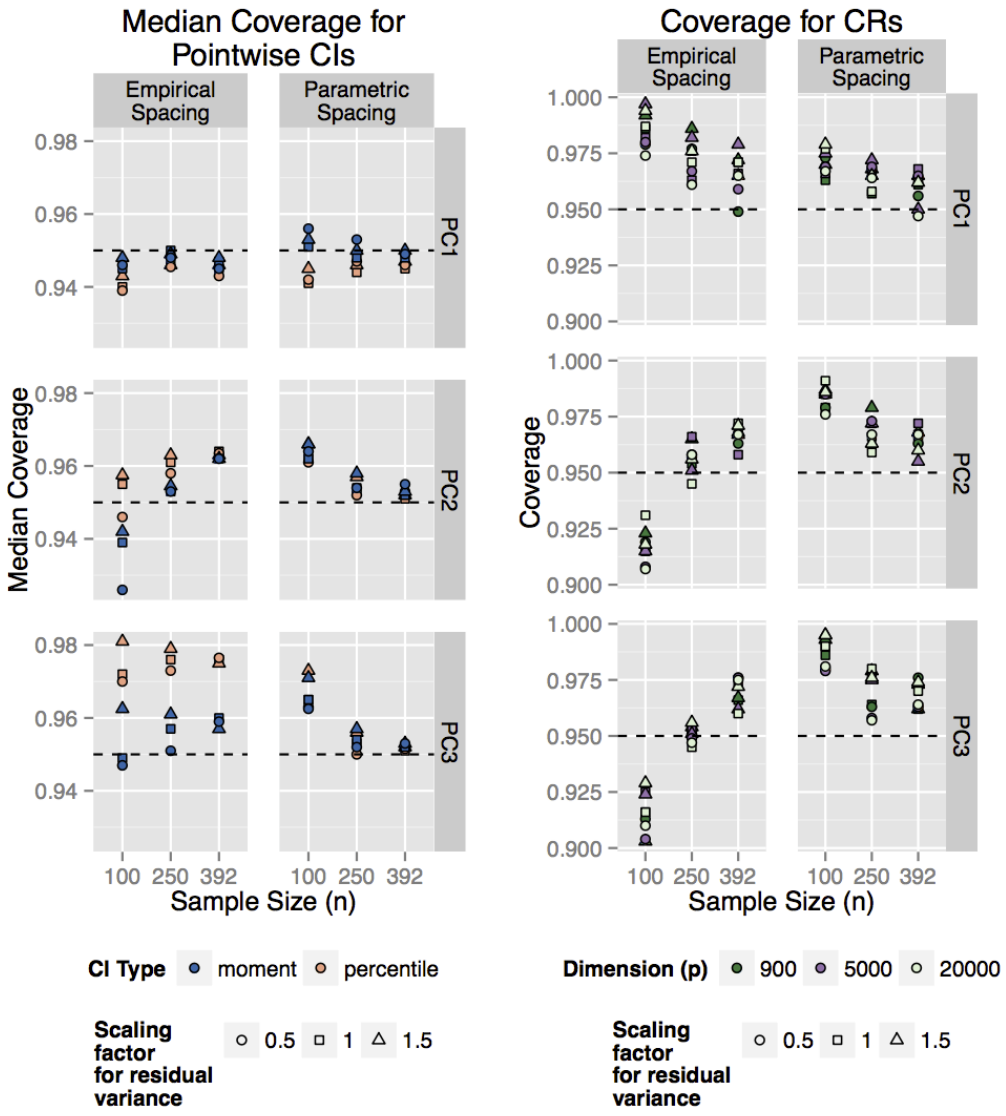

Figure 3.2: Coverage across simulation scenarios - The (3×2) array of plots on the left shows the median coverage rate across all p estimated CIs for the PC elements ($p = 900$). Rows correspond to the PC being estimated. Simulation cases using the empirical eigenvalue spacing are shown on the left column, and simulation cases where where each PC explains half as much as the previous PC are shown on the right column. The (3×2) array of plots on the right shows coverage for CRs for the PCs.

and PC3, the moment-based intervals give close to 95% coverage, but that the percentile intervals may give poor coverage in certain regions.

The right side of Figure 3.2 shows coverage rates of confidence cones for the principal components (section 3.3.3). Coverage appears to improve when the eigenvalues are well spaced and when sample size increases. Differences in coverage are also more noticeable here than in the median coverage rates for pointwise intervals. Coverage rates of CRs for the principal subspace (section 3.3.3) are shown in the supplemental materials, and follow the same general pattern as CRs for the individual PCs.

To more formally summarize our simulation results for the confidence cones, we modeled PC coverage rate as a function of the sample dimension, sample size, eigenvalue spacing, and residual noise variance. Specifically, we considered the ordinary linear regression model $|Coverage - .95| = \beta_0 + \beta_1 log(p) + \beta_2 n + \beta_3 s + \beta_4 f + e$, where s is an indicator of the parametric spacing for the eigenvalues, f is the scaling factor applied to the variance of the residual noise in the simulation, and e is a random normal error accounting for unmodeled variability in coverage. We separately fit this model on coverage rates for each PC, treating the all coverage rates as having independent and identically distributed random errors. For PC1, larger sample sizes and the parametric eigenvalue spacing both significantly improved coverage ($\hat{\beta}_2 = -5.1 \times 10^{-5}$, $\hat{\beta}_3 = .0097$, with 95% CIs: $(-6.5 \times 10^{-5}, -3.8 \times 10^{-5})$ and $(-0.013, -0.0064)$ respectively), and higher levels of residual noise significantly worsened coverage ($\hat{\beta}_4 = 0.0084$, 95% CI: $(0.0045, 0.012)$). For PC2 and PC3, larger sample sizes also significantly improved coverage ($\hat{\beta}_2$ estimates 6.1×10^{-5} and -6.1×10^{-5} respectively, with 95% CIs: $(-8.0 \times 10^{-5}, -4.2 \times 10^{-5})$ and $(-8.5 \times 10^{-5}, -3.5 \times 10^{-5})$), but no

other variables had significant effects.

We also studied coverage of parameters relevant to the first three score variables. Because the scores themselves are subject-specific random effects rather than population parameters, we focused on coverage of best linear unbiased predictors (BLUPs) for the score variables (Robinson, 1991). We calculated the true BLUPs conditional on the observed data matrix being equal to $\mathbf{\Psi S}$, where $\mathbf{S}$ is the matrix from which we draw the score variables $(s_{i1}, s_{i2}, \ldots, s_{iK_0})$, and $\mathbf{\Psi}$ is the matrix of first K_0 true population PCs. In each bootstrap sample we then calculated the empirical BLUPs (EBLUPs) for the scores (Fitzmaurice et al., 2012), and used the bootstrap distribution of the EBLUPs to calculate percentile and moment-based CIs.

Coverage rates for the BLUP CIs generally followed a similar pattern as coverage rates for the pointwise PC CIs, although the coverage was worse when the sample size was small and the residual noise was high. In the smallest sample size tested, coverage of BLUPs was as low as 85% coverage for the percentile CIs, and 90% for the moment-based CIs. Poorer coverage in these scenarios is to be expected though, as the EBLUPs depend not only on estimates of the PCs, but also on estimates of the eigenvalues of the population covariance matrix, which are known to be biased (Daudin et al., 1988). Note that if we had instead focused on estimates of $\mathbf{\Psi}' y_i$, then proper coverage would have been implied by proper coverage of the pointwise CIs for the PCs, as both parameters are projections of the true basis vectors. A full description of coverage rates for the BLUPs, as well as the calculation procedure for the BLUPs and EBLUPs, is given in the supplemental materials.

As a secondary analysis, we also looked at the distribution of the angles

between the sample PCs and the true population PCs. In general, when the k^{th} eigenvalue of the population covariance matrix was on a different order of magnitude than the other eigenvalues, the k^{th} sample PC tended to be close to the k^{th} population PC. This was the case for PC1 in the empirical spacing scenario, and PCs 1 through 3 in the parametric spacing scenario. When the leading five eigenvalues of the population covariance matrix were not well separated from each other but were well separated from the remaining eigenvalues, the individual sample PCs were not necessarily close to their corresponding population PCs but did tend to be close to the subspace spanned by the leading population PCs. This was the case for PCs 2 and 3 in the empirical spacing scenario. These results are all consistent with what we would expect based on Theorem 2 of Jung and Marron (2009). Because we fixed the proportion of variability explained by each PC, regardless of dimension, our increases in dimension correspond to the case described in Shen et al. (2012a) where the dimension and the leading eigenvalues all grow at the same rate. In this context, Theorem 4.1 of Shen et al. (2012a) suggests that our sample PCs should converge to their population counterparts as n increases, regardless of dimension. This is indeed what we see in our results (see the supplemental materials of this chapter).

3.5 Applying fast bootstrap PCA

3.5.1 Sleep EEG

When applying fast bootstrap PCA to the EEG dataset, we find that bootstrap estimates of PC1 exhibit minimal variability. PC2 and PC3 are estimated with

considerably more variability, but most of this variability is due to random rotations among PCs 2 through 4, all of which roughly correspond with oscillatory patterns.

Figure 3.3 shows the results of this analysis. The first row shows 95% pointwise intervals for each dimension of each of the three PCs. A random subsample of 30 draws from the bootstrap distribution of each PC are shown in gray. We see that the moment-based and percentile intervals generally agree, although they tend to differ more when the fitted PC elements are further from zero. Since the width of the percentile and moment-based CIs are fairly similar, disagreements between the two types of intervals are reflective of skewness in the underlying bootstrap distribution.

The sets of pointwise intervals shown in the top row of Figure 3.3 form bands around the fitted sample PCs. It's important to note these bands are only calibrated for pointwise 95% coverage – they are not expected to simultaneously contain the true population PC in 95% of samples. Statements about the overall shape of the population PCs that are based on these intervals will be somewhat ad hoc. Furthermore, many curves contained within these bands do not satisfy the norm 1 requirement for principal components, and are not valid solutions to PCA. For example, the upper and lower boundaries of the bands do not have norm 1, and thus are not in the parameter space for the PCs. Similarly, the zero vector is also not in the parameter space.

The top row of Figure 3.3 shows that both sets of intervals for PC1 are fairly tight, implying that there is little sampling variability in PC1. The pointwise CIs for PC2 are wider, especially in the first four hours of the night. If examined alone, this feature of the CIs might erroneously lead readers to think that the

oscillatory pattern in $\mathbf{V}_{[,2]}$ is artificial, and not present in the population PC. However, if we also look at a subsample of draws from the bootstrap distribution of PC2 (shown in gray), we see that the negative spike in hour 1 and the positive spike in hour 2 are often shifted in bootstrap samples. Pointwise variability in the oscillatory pattern is better explained by a simultaneous shift of both peaks than by a magnitude change in either peak. Those bootstrap draws of $\mathbf{V}^b_{[,2]}$ that are most shifted tend to bear a closer resemblance to $\mathbf{V}_{[,3]}$.

This resemblance is shown more directly in the bottom row of Figure 3.3, which shows pointwise CIs summarizing the distribution of $\mathbf{A}^b_{[,k]}$ for $k = 1, 2, 3$. Recall that the bootstrap PCs are equal to $\mathbf{V}^b = \mathbf{V}\mathbf{A}^b$, such that $\mathbf{A}^b_{[j,k]}$ represents the weight that the k^{th} PC of the b^{th} bootstrap sample ($\mathbf{V}^b_{[,k]}$) places on the j^{th} PC of the original sample ($\mathbf{V}_{[,j]}$). Low bootstrap variability for the k^{th} PC is generally characterized by $\mathbf{A}^b_{[k,k]}$ being close to 1, and all other elements of $\mathbf{A}^b_{[,k]}$ being close to zero. While this is the case for bootstrap variability in PC1 (bottom-left panel of Figure 3.3), the bootstrap draws of PC2 tend to place high weight on $\mathbf{V}_{[,3]}$, in addition to $\mathbf{V}_{[,2]}$. Equivalently put, bootstrap draws for both $\mathbf{A}^b_{[2,2]}$ and $\mathbf{A}^b_{[3,2]}$ tend to have high absolute values (bottom-center panel of Figure 3.3). A similar pattern is shown for PC3. Overall, the bottom row of Figure 3.3 shows that the majority of the variation in PCs 2-3 is due to rotations among the leading PCs.

Note that the moment-based CIs shown on the right column of Figure 3.3 can exceed one in absolute value, which will surely violate the norm condition for PCs. In practice, such violations should be accounted for by truncating the CIs at -1 and 1, but we keep the violation for illustrative purposes in Figure 3.3. It is also worth noting that the percentile CIs for $\mathbf{A}^b_{[k,k]}$ will rarely include

the value 1, which can be thought of as the fitted value of $\mathbf{A}^b_{[k,k]}$ in the original sample (shown in black in Figure 3.3). The low dimensional percentile CIs for the elements of $\mathbf{A}^b$ also fully contain the information required to create confidence cones for each PC (section 3.3.3).

Figure 3.4 shows the bootstrap distribution of the first three eigenvalues of the sample covariance matrix (the diagonals of $(1/(n-1))(\mathbf{D}^b)^2$). In general, there is a known upward bias in the first eigenvalue of the sample covariance matrix, relative to the first eigenvalue of the population covariance matrix (Daudin et al., 1988). The amount of bias can be estimated using bias in the bootstrap distribution of covariance matrix eigenvalues. Each bootstrap sample can be seen as a simulated draw from the original sample, in which the eigenvalues are known. Here, we define the percent bias in the bootstrap eigenvalues as the difference between the average eigenvalue across all bootstrap samples and the eigenvalue in the original sample, divided by the eigenvalue of the original sample. For the first three covariance matrix eigenvalues in the EEG dataset (Figure 3.4), there is only a slight upward bias in the bootstrap estimates (percent bias = 1.1%, 4.5%, and 5.0% respectively).

3.5.2 Brain MRIs

We also apply our bootstrap procedure to estimate sampling variability of the PCs from the brain MRI dataset. This is primarily included as an example to show the computational feasibility of our method in the high dimensional setting. A deeper interpretation of the sample PCs is provided by (Zipunnikov et al., 2011b,a).

Our results imply that PC1 is estimated with fairly low sampling variability,

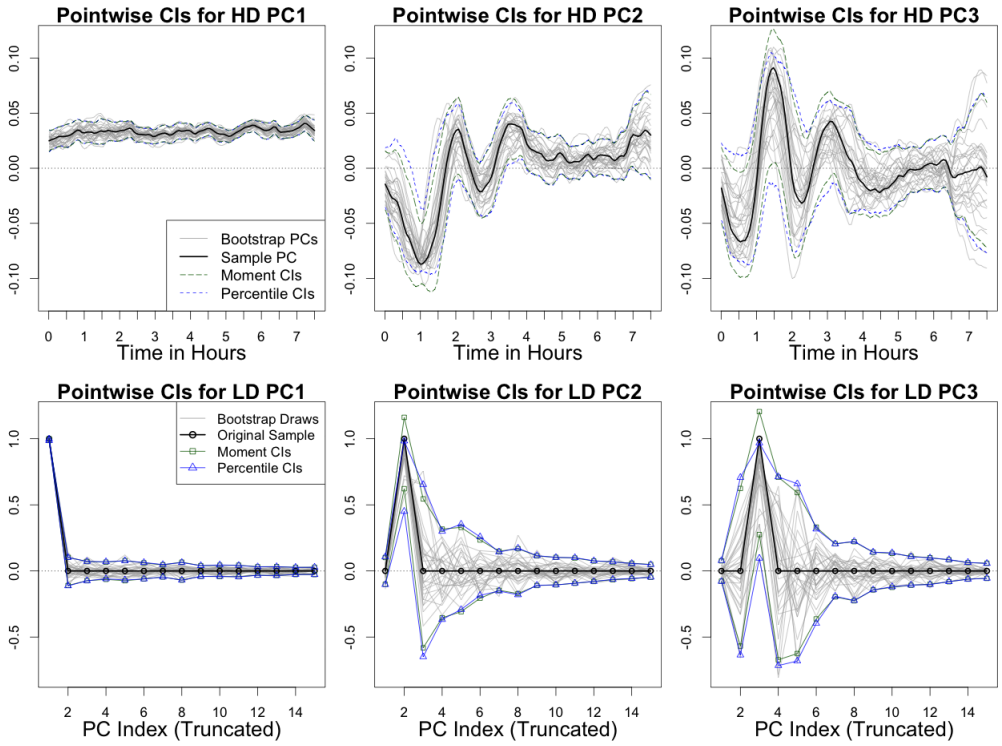

Figure 3.3: Bootstrap PC variability - Each column of plots corresponds to a different PC, either the first, second or third. The top row shows the fitted principal components on the original high dimensional space ($\mathbf{V}_{[,k]}$ for $k = 1, 2, 3$), along with pointwise confidence intervals, and 30 draws from the bootstrap distribution. The bottom row shows the same information, but for the low dimensional representation of the bootstrap PCs ($\mathbf{A}^b_{[,k]}$ for $k = 1, 2, 3$). In the bottom row, the thick black line corresponds to the case when $\mathbf{A}^b_{[,k]} = \mathbf{I}_{n[,k]}$, where $\mathbf{I}_{n[,k]}$ is the k^{th} column of the $n \times n$ identity matrix, such that $\mathbf{V}^b_{[,k]} = \mathbf{V}\mathbf{A}^b_{[,k]} = \mathbf{V}_{[,k]}$.

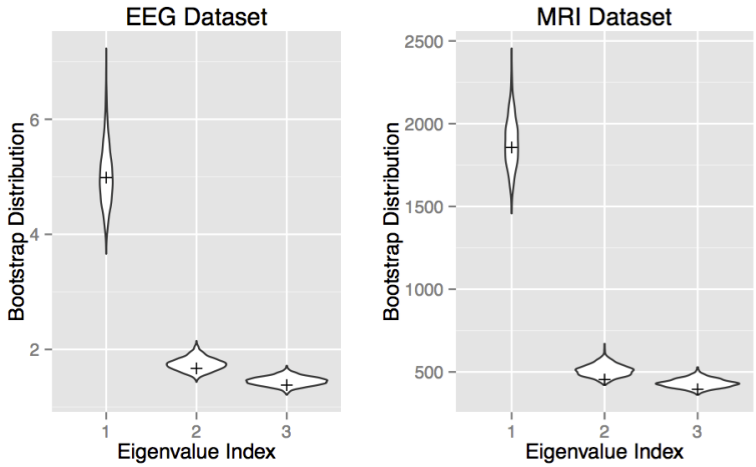

Figure 3.4: Bootstrap eigenvalue distribution - For both the EEG and MRI datasets, we show bootstrap distribution for the first three eigenvalues of the sample covariance matrix. Tick marks show the eigenvalues from the original sample covariance matrix.

but that sampling variability is higher for PC2 and PC3. The first two rows of Figure 3.5 respectively show the fitted sample PCs and the bootstrap standard errors for the PCs. For PC1, the standard errors are generally of a lower order of magnitude than the corresponding fitted values. A direct comparison is given in the bottom row of Figure 3.5, which shows the fitted sample PCs divided by their pointwise bootstrap standard errors. These ratios can be interpreted as Z-scores under the element-wise null hypotheses that the value of any one element of the population PC is zero. Z-scores with absolute value less than 1.96 are omitted from the display.

To estimate sampling variability due to rotations of the leading population PCs, Figure 3.6 shows pointwise confidence intervals for the truncated vectors $\mathbf{A}^b_{[,k]}$, for $k = 1, 2, 3$. These intervals are analogous to the intervals shown in the bottom row of Figure 3.3. A substantial proportion of the bootstrap variability

for the second two PCs is due to random rotations between them.

The second panel of Figure 3.4 shows the bootstrap distribution of the eigenvalues of sample covariance matrix. Relative to the fitted eigenvalues in the original sample, the bootstrap eigenvalues show a small, but notable upward bias (percent bias = 1.7%, 12.2%, and 9.2% respectively). Figure 3.4 also illustrates that, for both datasets, the first eigenvalue is well separated from the second and third eigenvalues. This makes the relatively small variability in PC1, and the largely rotational variability in PCs 2 and 3, consistent with what we would expect from Theorem 2 of Jung and Marron (2009).

3.6 Discussion

In this paper we outline methods for fast PCA in high dimensional bootstrap samples, based on the fact that all bootstrap samples lie in the same low dimensional subspace. We show computational feasibility by applying this method to a sample of sleep EEG recordings ($p = 900$), and to a sample of processed brain MRIs (p =2,979,666). Bootstrap standard errors for the first three components of the MRI dataset were calculated on a commercial laptop in 47 minutes, as opposed to approximately 4 days with standard methods (see supplemental materials for computational comparisons against standard methods for different values of p and n).

Ultimately, the usefulness of high dimensional bootstrap PCA will depend not on its speed, but on its demonstrated ability to capture sampling variability. We found that the bootstrap performed well in the simulation settings presented here (section 3.4). However, bootstrap PCA has rarely been applied to high

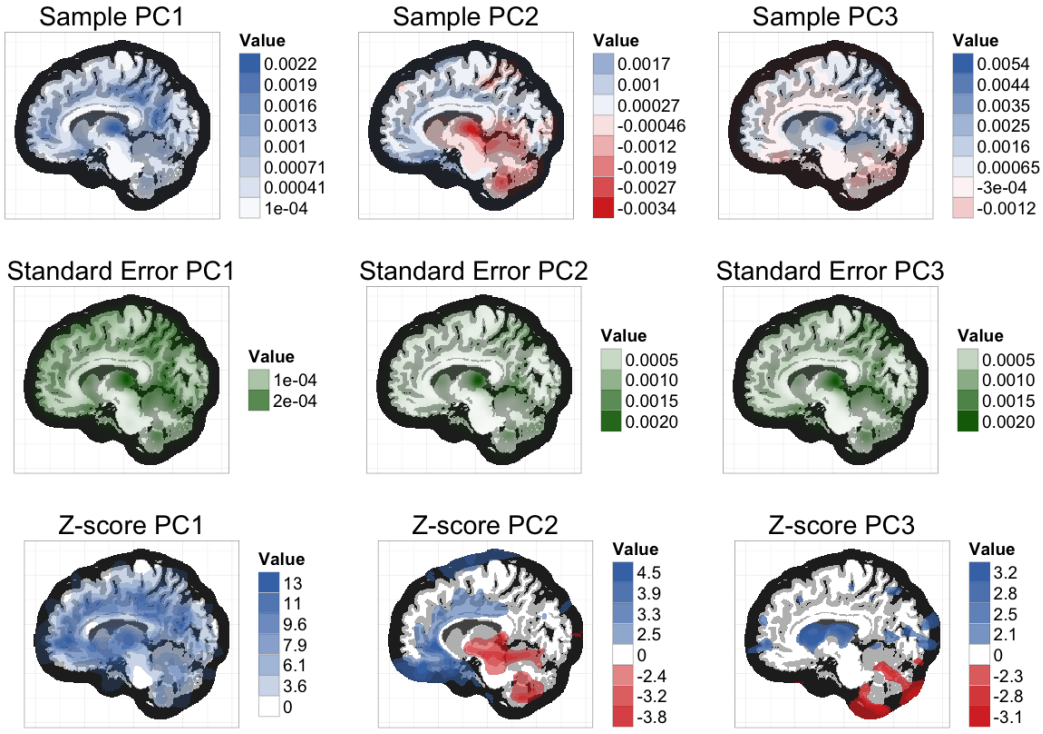

Figure 3.5: Fitted sample values, bootstrap standard errors, and Z-scores for the MRI PCs - The voxelwise values for the PCs and Z-scores (top and bottom rows) have been binned, and shaded according to the value of their corresponding bin's midpoint. This allows us to visually show both sign (color) and magnitude (opacity). Because the standard errors (middle row) are always positive, the binning procedure is not necessary, and the voxels are shaded on a continuous scale.

dimensional data in the past, and its theoretical properties in high dimensions are still not well understood. Specifically, to our knowledge, the theoretical coverage of bootstrap-based confidence intervals have not been well studied. The lack of study on this topic is likely due, in part, to the computational bottlenecks of standard bootstrap PCA, which are compounded in theoretical research that includes simulation studies. Our hope is that the methods presented here will expand the use of bootstrap PCA, and allow for theoretical properties of the

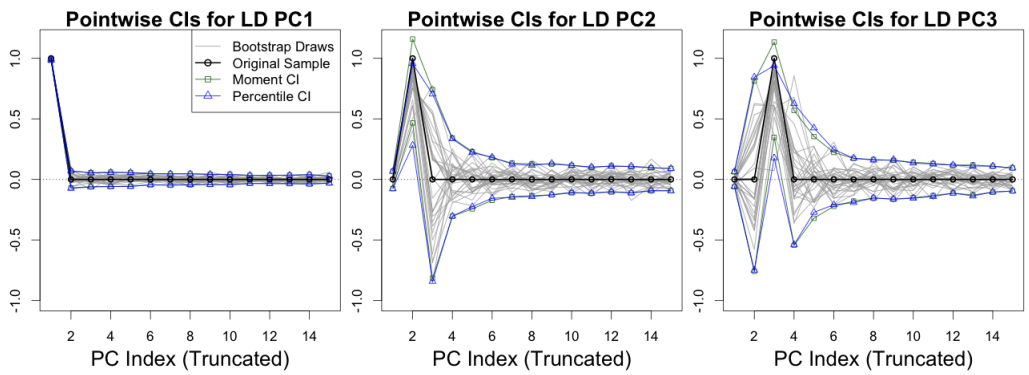

Figure 3.6: Low dimensional CIs for the MRI PCs - Moment-based CIs, percentile CIs, and 30 random bootstrap draws for $\mathbf{A}^b_{[1:15,k]}$, where $k = 1, 2$ and 3.

bootstrap PCA procedure to be studied and verified via simulations.

When interpreting the results of bootstrap PCA, we find it particularly useful to generate confidence intervals around elements of the low dimensional $\mathbf{A}^b$ matrices (Figures 3.3 and 3.6). These CIs are a parsimonious way to display the dominant directions in PC bootstrap variability, which often correspond to rotations among the leading sample PCs. Calculating these CIs also does not require operations on the p-dimensional scale, beyond the initial SVD of the sample. Another potential way to summarize the dominant directions of PC bootstrap variability would be to create elliptical CRs constrained to the p-dimensional hypersphere, a topic which we discuss in the supplementary materials.

Interpretation of bootstrap PCA results is complicated by the fact that many PCA results are interdependent. For example, each PC is only defined conditionally on the preceding PCs. If we want to isolate only the variability of the k^{th} PC that affects this conditional interpretation, it can be useful to first assume that the first $k-1$ PCs are estimated without error. Logistically, we can condition on the leading $k-1$ PCs by resampling from the residuals after

projecting the dataset onto the matrix $\mathbf{V}_{[,1:(k-1)]}$. This is equivalent to setting the first $k-1$ score variables to zero before starting the resampling process. Alternatively, we could assume that the first PC is a mean shift, and estimate the sampling variability of the remaining PCs by resampling from the residuals after projecting the dataset onto a constant, flat basis vector. This general approach requires the strong assumption that the leading PCs are known, but the procedure can still be useful in exploring the sources of PC variability.

R Package Code

Code for this paper is available as an R package at

`http://cran.r-project.org/web/packages/bootSVD/index.html`

Acknowledgements

This chapter is joint work with Brian Caffo, Brian Schwartz, and Vadim Zipunnikov. It has been accepted into the Journal of the American Statistical Associate (Theory & Methods) and will appear shortly.

Chapter 4

Stochastic optimization of adaptive enrichment designs for two subpopulations

4.1 Introduction

Prior uncertainty regarding treatment effect heterogeneity can pose a challenge to trial designers. If the treatment only benefits a subset of the population, standard clinical trials enrolling from the entire population may have low power. On the other hand, if the entire population benefits, a standard trial enrolling only one subpopulation will not provide any information about the complementary population.

These issues can be mitigated with the use of an adaptive enrichment trial design. Such designs consist of a set of decision rules for early stopping of participant enrollment in different population subsets based on interim analyses of accrued data at predefined stages (Wang et al., 2009). For example, early stopping can occur if there is strong evidence early in the trial of the treatment's benefit or harm for a subpopulation. The design also includes a procedure to test null hypotheses for each population of interest. Alternatively, one-stage,

non-adaptive designs that test hypotheses on multiple subpopulations can also be constructed. One-stage designs will often have a lower maximum sample size than adaptive designs, but at the cost of a higher expected sample size due to their lack of ability to stop early.

We aim to optimize the enrollment modification rule and multiple testing procedure for an adaptive enrichment design. The goal is to minimize either expected sample size or expected trial duration, under constraints on power and the Type I error rate. We focus on designs that are guaranteed to strongly control the familywise Type I error rate, i.e., where the probability is at most α that one or more true null hypotheses is rejected, regardless of the (unknown) data generating distribution.

The optimization problems we consider are challenging in that no existing approach is guaranteed to find the global optimum. The main difficulty is that there are many design parameters to optimize over, as well as many constraints. The parameters in our adaptive enrichment designs include the following (plus additional parameters in some settings): the number of stages; per-stage sample sizes; and, for each population and stage, an efficacy boundary and futility boundary. As an example, in the case of 2 subpopulations and 5 stages, there are over 30 variation independent parameters. We consider searches that do not restrict the parameter space by imposing a preset structure, e.g., forcing the boundaries to be proportional to those of O'Brien and Fleming (1979). To the best of our knowledge, we are the first to address the problem of optimizing adaptive enrichment designs with such a large number of parameters. Additionally, we consider using the results of an unstructured search to inform a choice of structure to use in a restricted parameter search.

While our approach based on simulated annealing (described below) does not ensure that a global optimum is found, we show that it can substantially reduce the expected sample size compared to standard designs. In one of the examples considered here, the proposed procedure reduces expected sample size by approximately 37%. However, in another example we consider, with longer follow-up time to measurement of patient outcomes, all adaptive designs performed poorly, as many patients must be enrolled before the first measurements are taken. Here the proposed procedure outperformed standard adaptive designs, but only improved the expected sample size of one-stage trials by less than 2%.

General approaches exist for constructing optimal designs for simpler problems, such as those involving a single null hypothesis (Eales and Jennison, 1992; Hampson and Jennison, 2013). Hampson and Jennison (2015) extend this approach to handle multiple hypotheses, but the resulting designs may not strongly control the familywise Type I error rate. Thall et al. (1988) perform a 2-dimensional grid search to minimize the expected sample size of a 2-stage trial comparing the effects of several treatments. Krisam and Kieser (2015) and Graf et al. (2015) consider optimizing different parametrization of a two-stage design with two subpopulations, and respectively search over 2-dimensional and 3-dimensional parameter spaces. In contrast, our aim is to search over more flexible, higher dimensional families of designs. For trials involving two subpopulations, optimal 2-stage designs can be found via sparse linear programming (Rosenblum et al., 2014), but this approach becomes computationally infeasible for more than two stages. The approach of Rosenblum et al. (2015) is restricted to O'Brien-Fleming boundaries, two null hypotheses, and a much smaller design

space than considered here.

For trial design problems where no existing approach is guaranteed to find an optimal solution, one may turn to general-purpose, approximate methods such as simulated annealing (SA). Wason et al. (2012) apply SA to optimize the estimated worst-case expected sample size of a group sequential design, with penalties added to the objective function for violations of either Type I or Type II error constraints. Wason and Jaki (2012) extend these results by applying SA to optimize a multi-arm, multi-stage trial where several treatments groups are compared against a shared, single control group.

Our optimization problem differs from that of Wason and Jaki (2012) in that our futility boundaries are non-binding (which is typically preferred by regulators such as the U.S. Food and Drug Administration, as noted by Liu and Anderson (2008)), and our designs allow continuation after one null hypothesis is rejected (so other hypotheses may be rejected at later stages). We also set our efficacy boundaries using error-spending functions in order to handle unknown information increments, and include a final adjustment step after the SA procedure to ensure that power constraints are met. Without such additions, optimizing a penalized objective function does not guarantee that Type I or Type II constraints will be satisfied. Another difference is that we apply a parallelized version of SA. These and other differences between our implementation of SA and that of Wason and Jaki (2012) are discussed in Section 4.4.

In Section 4.2, we introduce motivating data examples based on a new surgical intervention for stroke, and on a hypothetical intervention for Alzheimer's disease. In Section 4.3, we introduce a class of adaptive enrichment designs, referred to hereafter as "adaptive designs." We discuss how efficacy boundaries

can be constructed by incorporating either the covariance of the test statistics (Rosenblum et al., 2015), or by using alpha-reallocation (Maurer and Bretz, 2013). We also introduce different levels of trial design complexity, which balance design flexibility versus simplicity. In Section 4.4, we outline our approach for optimization. In Section 4.5, we explore the performance of each type of trial, and compare to standard trial designs using approximate O'Brien Fleming boundaries (O'Brien and Fleming, 1979) or Pocock boundaries (Pocock, 1977). We end with a discussion of future work.

4.2 Applications

4.2.1 Application 1: surgical treatment of stroke (MISTIE)

We first describe an example of planning a Phase III trial of a surgical treatment for stroke, which was also considered by Rosenblum et al. (2015). The treatment is called Minimally-Invasive Surgery Plus rt-PA for Intracerebral Hemorrhage (MISTIE), and is described in detail by Morgan et al. (2008). The primary outcome is based on each participant's disability score on the modified Rankin Scale (mRS) measured 180 days (d) from enrollment. A successful outcome is defined as a mRS score less than or equal to 3.

In planning the Phase III trial, the investigators were interested in two subpopulations defined by size of intraventricular hemorrhage (IVH) at baseline. "Small IVH" participants are defined to have IVH volume less than 10ml and not requiring a catheter for intracranial pressure monitoring. The remaining participants are called "large IVH". The Phase II trial only recruited small IVH participants. A preliminary analysis of the data resulted in an estimated

treatment effect of approximately 12.1%. Knowledge of the underlying biology of these types of brain hemorrhage suggested a possible benefit for those with large IVH as well. However, there was greater uncertainty about the treatment effect in the large IVH subpopulation. Investigators inquired about the possibility of running a phase III trial that included both small IVH and large IVH participants, but with the option to stop a subpopulation's enrollent (using a preplanned rule) if interim data indicated that a benefit was unlikely.

4.2.2 Application 2: Alzheimer's Disease Neuroimaging Initiative (ADNI)

We also consider an example based on data from the Alzheimer's Disease Neuroimaging Initiative (ADNI, www.adni-info.org), which prospectively follows a cohort with mild cognitive impairment or early Alzheimer's disease at baseline. The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging, positron emission tomography, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment and early Alzheimer's disease. We focus on subpopulations defined by a participant's apolipoprotein E (APOE) $\epsilon 4$ allele carrier status, which is associated with increased risk of late onset Alzheimer's disease (Sadigh-Eteghad et al., 2012).

Clinical investigators, who were planning a Phase III trial of a new treatment to prevent progression from mild cognitive impairment to Alzheimer's disease, suspected that there may be treatment effect heterogeneity across carrier status.

The primary outcome is the 2-year change in Clinical Dementia Rating Sum of Boxes score (CDR-SB), an aggregate measure of Alzheimer's symptom severity. The enrollment rate was projected to be approximately 500 participants per year.

4.3 Adaptive trial designs

4.3.1 Notation, hypotheses, and test statistics

We consider two subpopulations that partition the overall population. Let $j = 1, 2, C$ be a group index respectively denoting subpopulation 1, subpopulation 2, or the combined population. The treatment effect for a population is defined as the difference between the mean outcome for treatment and control. Outcomes may be continuous, binary, or on any scale that allows the treatment effect to be estimated with a difference in means z-statistic. The data collected for each participant is a vector (S, A, Y) representing his/her subpopulation, study arm assignment, and outcome, respectively. We assume each participant's data vector is an independent, identically distributed draw from an unknown joint distribution on (S, A, Y).

Let π_j denote the proportion of the combined population in group j (with $\pi_C = 1$ by convention). Let δ_j denote the treatment effect in group j. It follows that $\delta_C = \pi_1\delta_1 + \pi_2\delta_2$. Let H_1, H_2, and H_C respectively be the null hypotheses of no average treatment benefit in groups 1, 2 and C, i.e., $H_j : \delta_j \leq 0$. Each corresponding alternative hypothesis has the form $\delta_j > 0$. Let σ^2_{tj} and σ^2_{cj} denote the variances in group j under treatment and control, respectively, with all variances assumed known. We assume that treatment assignment is randomized

with probability $\frac{1}{2}$, and that each participant's outcome is measured after a delay of d years from enrollment.

Our designs have $K > 0$ stages, each concluding with an analysis. These analyses may lead to stopping enrollment in one or both subpopulations, according to a set of predefined rules that are functions of the accrued data. The k^{th} stage is said to be completed once $\pi_j n_k$ additional participant outcomes have been measured from each subpopulation j that is still being enrolled. The n_k terms are predetermined design parameters. Let $N := \sum_{k=1}^{K} n_k$ denote the maximum total sample size. The enrollment decision at the end of stage k takes as input the following cumulative test-statistic:

$$Z_j^{(k)} := \hat{\delta}_j \left(\frac{\sigma_{cj}^2 + \sigma_{tj}^2}{\frac{1}{2} \sum_{k'=1}^{k} n_{k'} \pi_j} \right)^{-\frac{1}{2}},$$

for each population $j \in \{1, 2, C\}$ that is still being enrolled, where $\pi_C = 1$, and $\hat{\delta}_j$ denotes the difference in sample means estimator for δ_j based on the accrued data. The combined population statistic $Z_C^{(k)}$ is undefined if one or more subpopulations had enrollment stopped early at a previous stage.

At the analysis just after stage k, each statistic $Z_j^{(k)}$ is compared against a predetermined efficacy boundary $e_j^{(k)}$ and futility boundary $f_j^{(k)}$. If $Z_j^{(k)} > e_j^{(k)}$, H_j is rejected. Equivalently, efficacy boundaries can also specified on the p-value space rather than the z-statistic space. Whenever H_1 and H_2 are both rejected, we automatically reject H_C as well. Rejecting any hypothesis implies that the hypothesis remains rejected in all future stages. If $Z_j^{(k)} \leq f_j^{(k)}$, then recruitment in group j is stopped for futility. We use non-binding futility boundaries, i.e., strong control of the familywise Type I error rate is asymptotically guaranteed for all designs in this paper, even if futility boundaries are ignored and both

subpopulations are enrolled through the end of the last stage K.

For each subpopulation $j \in \{1, 2\}$, its enrollment continues until one (or more) of the following occurs: H_j is rejected ($Z_j^{(k)} > e_j^{(k)}$), subpopulation j is stopped for futility ($Z_j^{(k)} \leq f_j^{(k)}$), the combined population is stopped for futility ($Z_C^{(k)} \leq f_C^{(k)}$), or the maximum sample size for subpopulation j is reached. Under this setup, rejecting the combined population null hypothesis H_C does not imply stopping all trial enrollment; further tests of H_1 and H_2 may still be conducted. Since $Z_C^{(k)}$ is only defined if the combined population is enrolled through stage k, we do not conduct the test $Z_C^{(k)} > e_C^{(k)}$ at stages after enrollment for at least one subpopulation has stopped; it is still possible to reject H_C at future stages, if both subpopulation null hypotheses are rejected.

Because participant outcomes are measured with delay, there can be enrolled participants at interim analyses whose outcomes are not yet measured and cannot be analyzed. These participants are referred to as "pipeline participants." Pipeline participants do not contribute to test statistic calculations, although they always contribute to realized sample size of the trial.

We compare two methods for calculating efficacy boundaries. The first uses an error spending approach based on the covariance matrix for the test statistics $Z_j^{(k)}$ (Rosenblum et al., 2015). The second method uses efficacy boundaries where the proportion of Type I error allocated to hypothesis H_j can be reallocated to remaining hypotheses if H_j is rejected (Bretz et al., 2009; Maurer and Bretz, 2013). We refer to these two types of multiple testing procedures as $\mathscr{H}^{COV}$ and $\mathscr{H}^{MB}$, respectively. For both, Type I error is calculated under the assumption that futility boundaries are never adhered to (to give the worst-case Type I error under non-binding futility boundaries). Power, expected sample

size, and expected duration are calculated under the assumption that futility boundaries are adhered to.

4.3.2 Multiple testing procedure 1: covariance approach

Rosenblum et al. (2015) propose a method for efficacy boundary calculation that incorporates the correlation between the test statistics (which is assumed known), both across stages and across groups. We refer to this approach as $\mathscr{H}^{COV}$. Rosenblum et al. (2015) show that for their designs to strongly control the familywise Type I error rate, it is sufficient to control the familywise Type I error rate under the global null hypothesis of no treatment effect in any subpopulation. Under this global null hypothesis the z-statistics described in Section 4.3.1 asymptotically follow a multivariate normal distribution with mean zero and covariance structure described by

$$Cov(Z_j^{(k)}, Z_j^{(l)}) = \sqrt{\frac{\sum_{k'=1}^{k} n_{k'}}{\sum_{l'=1}^{l} n_{l'}}}, \qquad Cov(Z_1^{(k)}, Z_2^{(l)}) = 0, \text{ and}$$

$$Cov(Z_j^{(k)}, Z_C^{(l)}) = \sqrt{\pi_j \left(\frac{\sigma_{cj}^2 + \sigma_{tj}^2}{\sigma_{cC}^2 + \sigma_{tC}^2}\right) \left(\frac{\sum_{k'=1}^{k} n_{k'}}{\sum_{l'=1}^{l} n_{l'}}\right)},$$

for $j = 1, 2, C$, and stages k and l such that $1 \leq k \leq l \leq K$.

Given this null distribution, the familywise Type I error can be controlled using an error spending approach. The first step of this approach is to pre-specify an ordering for the hypothesis tests. For example, pre-specifying the ordering $1 < 2 < C$ implies that we always first test H_1, then H_2, and finally H_C, at each stage where all three hypotheses are still being tested. If only a subset

of hypotheses are still being tested at a given stage, the same ordering applies to these remaining hypotheses. The second step is to limit the probability that any one test statistic $Z_j^{(k)}$ is the first test statistic to lead to a rejection of any hypothesis. Let $\alpha_j^{(k)}$ be this allowed probability that $Z_j^{(k)}$ is the first test statistic to lead to a rejection. Let α denote the required familywise Type I error rate, with $\sum_{j\in\{1,2,C\}}\sum_{k=1}^{K}\alpha_j^{(k)} = \alpha$. Finally, let $w_j := \frac{\sum_{k=1}^{K}\alpha_j^{(k)}}{\alpha}$ be the proportion of α allocated to hypothesis H_j. The efficacy boundaries $\{e_j^{(k)}\}_{(j,k)\in\{(1,2,C)\times(1:K)\}}$ can then be iteratively calculated by solving

$$P_{H_1\cap H_2}(Z_j^{(k)} > e_j^{(k)};\ \text{and } Z_{j'}^{(k')} \leq e_{j'}^{(k')} \text{ for all } k', j' \text{ such that } (k', j') \prec (k, j)) = \alpha_j^{(k)}, \tag{4.1}$$

using the known covariance for the test statistics. Here, we define the condition $(k', j') \prec (k, j)$ to hold whenever $k' < k$, or when ($k' = k$ and $j' < j$).

Rather than individually specifying each $\alpha_j^{(k)}$, it is common to instead specify a structured alpha spending function to determine the alpha allocation across stages. For example, we consider setting the alpha allocated to each stage k and hypothesis H_j to be $w_j \times (a_k - a_{k-1})$, where $a_k = \left(\frac{\sum_{k'=1}^{k} n_{k'}}{N}\right)^{\rho_{ej}}$ and $a_0 = 0$. In this way, rather than choosing $3 \times K$ values for the $\alpha_j^{(k)}$ terms, investigators need only specify the six design parameters w_j and ρ_{ej} for $j = 1, 2, C$. For a group sequential design testing only one hypothesis, setting ρ_{ej} to be 1 or 3 results efficacy boundaries that are similar to Pocock boundaries (Pocock, 1977) or O'Brien Fleming boundaries (O'Brien and Fleming, 1979) respectively (Jennison and Turnbull, 1999).

4.3.3 Multiple testing procedure 2: alpha-reallocation approach

Maurer and Bretz (2013) propose a procedure that allows for alpha-reallocation if one of the hypotheses is rejected, which we refer to here as $\mathscr{H}^{MB}$. This approach also accounts for correlation of test statistics across stages, but does not explicitly adjust for correlation of test statistics across hypotheses.

Before the start of the trial, investigators initialize a set of Type I error weights w_j for $j = 1, 2, C$ such that $\sum_j w_j = 1$ and αw_j is the allowed probability under the global null that H_j is rejected. As described in more detail below, these weights can later be changed if one or more hypotheses is rejected during the trial. For each hypothesis H_j, investigators also choose a set of nonnegative functions α_{kj} which further subdivide the error rate αw_j across stages of the trial. Specifically, $\alpha_{jk}(\alpha w_j)$ represents the portion of αw_j to be allocated to each stage k, or the probability under the global null that H_j is rejected in stage k. As in Section 4.3.2, investigators may choose to specify a structured alpha spending function for each hypothesis rather than specifying separate α_{jk} functions for each combination of j and k.

Test-statistics are constructed by converting each $Z_j^{(k)}$ into a p-value $p_j^{(k)}$. Each p-value $p_j^{(k)}$ is then compared against a nominal boundary denoted by $\alpha^*_{jk}(\alpha w_j)$, which is calibrated to produce the appropriate Type I error rate of $\alpha_{jk}(\alpha w_j)$. The value $\alpha^*_{jk}(\alpha w_j)$ generally will not equal $\alpha_{jk}(\alpha w_j)$ unless a Bonferroni correction is used to adjust for multiple tests across stages (i.e. tests of $Z_j^{(k)}$ and $Z_j^{(k')}$). Instead, a more powerful test procedure results from calibrating $\alpha^*_{jk}(\alpha w_j)$ using the known correlation across stages (i.e. $Cov(Z_j^{(k)}, Z_j^{(k')})$). At

each stage, the hypotheses H_j is rejected if $p_j^{(k)} < \alpha_{jk}^*(\alpha w_j)$.

If any hypothesis is rejected, the weights for the other hypothesis may be proportionally increased according to a pre-specified procedure. This reallocation allows for increased power in the testing procedure, despite a lack of explicit adjustment for correlations across hypotheses. Weight reallocations must be pre-specified as transition matrix, which can be intuitively visualized as a graph (Bretz et al., 2009). Specifically, let g_{ij} be the proportion of w_i to be reallocated to H_j in the event that H_i is rejected. If H_i is rejected we calculate an updated weight $w'_j = w_j + w_i g_{ij}$, and then calculate new boundaries for H_j equal to $\alpha_{jk}^*(\alpha w'_j)$ for each remaining hypothesis not yet rejected. From this point onward, each hypothesis H_j can be tested using the higher, less conservative boundaries $\alpha_{jk}^*(\alpha w'_j)$. The transition weights g_{ij} must also be adjusted after each rejection, to reflect the fact that a hypothesis has been removed. Every additional hypothesis rejection results in an additional reallocation of weights. For example, in our case of three hypotheses, rejecting two hypothesis results in a weight of 1 for the remaining hypothesis in the remaining stages of the trial.

In this way, rejecting any one hypothesis gives us greater power to reject other hypotheses without inflating the familywise Type I error. One intuition for this is that familywise Type I error does not reflect the number of false rejections, only whether any false rejections have occurred. Thus, once a hypothesis is rejected, it no longer remains necessary to continue correcting for that hypothesis. Maurer and Bretz (2013) prove that this procedure is equivalent to a consonant closed testing procedure, and therefore strongly controls Type I error.

4.4 Optimization

4.4.1 Power constraints and goals for optimization

Let $\underline{\delta} = (\delta^{(1)}, \delta^{(2)})$ denote a vector of possible values for the treatment effect in each subpopulation, and let $\delta^{\min} > 0$ denote the minimum value of the treatment effect that is clinically meaningful. We consider the following values for $\underline{\delta}$:

$$\underline{\delta}^{(0)} = (0,0); \qquad \underline{\delta}^{(1)} = (\delta^{min}, 0); \qquad \underline{\delta}^{(2)} = (0, \delta^{min}); \qquad \text{and} \qquad \underline{\delta}^{(C)} = (\delta^{min}, \delta^{min}).$$

Let $\mathscr{D}$ be a trial design, which contains a list of values for the parameters necessary to fully specify the analysis plan for a trial (i.e. K, $\{n_k\}_{k=1}^K$, initial alpha allocations, alpha reallocation rules, futility boundaries, and a hypothesis testing framework such as $\mathscr{H}^{MB}$ or $\mathscr{H}^{COV}$). Let $1 - \beta_j(\underline{\delta}', \mathscr{D})$ be the power of the design $\mathscr{D}$ to reject at least H_j by the end of the trial when (δ_1, δ_2) is equal to the vector $\underline{\delta}'$. We consider the following constraints on power and familywise Type I error:

1. $1 - \beta_1(\underline{\delta}^{(1)}, \mathscr{D}) \geq 0.8$
2. $1 - \beta_2(\underline{\delta}^{(2)}, \mathscr{D}) \geq 0.8$
3. $1 - \beta_C(\underline{\delta}^{(C)}, \mathscr{D}) \geq 0.8$
4. Strong control of the familywise Type I error rate, i.e.,

$$\sup_{\delta_1, \delta_2 \in \mathbb{R}} P_{\delta_1, \delta_2}(\text{reject one or more true null hypotheses}) \leq \alpha = 0.025.$$

Expected sample size is computed with respect to a distribution Λ on the possible treatment effects (δ_1, δ_2). We refer to this as the prior distribution on

the treatment effects. However, all of our designs have guaranteed asymptotic, familywise Type I error control without regard to this prior, i.e., it holds for any possible pair (δ_1, δ_2).

Subject to the constraints above, we aim to minimize the expected sample size averaged over Λ. In other words, subject to the above constraints, we aim to minimize over $\mathscr{D}$:

$$E_\Lambda(\tilde{n}(\mathscr{D})) := \int_{\delta_1,\delta_2} E_{\delta_1,\delta_2}(\text{total participants enrolled}) d\Lambda(\delta_1, \delta_2), \tag{4.2}$$

where $\tilde{n}(\mathscr{D})$ is a random variable denoting the realized sample size from the trial design $\mathscr{D}$. In this paper, we set Λ to be a discrete distribution with equal mass at $\underline{\delta}^{(0)}, \underline{\delta}^{(1)}, \underline{\delta}^{(2)}$, and $\underline{\delta}^{(C)}$. Under such a prior, $E_\Lambda(\tilde{n}(\mathscr{D}))$ is the average expected sample size across these four scenarios. While minimizing expected sample size is our primary goal, we also consider the problem of minimizing expected duration – the expected time from start of enrollment until both subpopulations have accrual stopped. This expectation is taken with respect to the same prior for the treatment effects. We combine our power constraints with the above expected sample size objective in Section 4.4.2.

4.4.2 Objective function

Due to the difficulty in directly solving the optimization problem (4.2) under the power and Type I error constraints, we instead define an unconstrained optimization problem where the constraints are incorporated as penalty terms as in (Wason and Jaki, 2012; Wason et al., 2012). The unconstrained objective function we aim to minimize over $\mathscr{D}$ is

$$J(\mathscr{D}) := E_\Lambda(\tilde{n}(\mathscr{D})) + \lambda \sum_{j\in\{1,2,C\}} \left(0.8 - (1 - \beta_j(\underline{\delta}^{(j)}, \mathscr{D}))\right)_+^3, \qquad (4.3)$$

where λ is a positive tuning parameter (set here to 100), and $(x)_+ = \max\{x, 0\}$. The first term can also be replaced with the expected trial duration. If any of the power constraints in Section 4.4.1 are violated, the objective function will incur a severe penalty. The exponent in the penalty term is meant to allow second order differentiability of $J(\mathscr{D})$ with respect to the power of the trial. This exponent is not necessary, but is potentially useful for some of the approaches discussed in Section 4.5.

Evaluating $J(\mathscr{D})$ requires the calculation of several multidimensional integrals. Due to the computational obstacle of these calculations, we instead estimate $J(\mathscr{D})$ via simulation. We used 10,000 simulation iterations, such that the Monte Carlo standard error for estimating a power close to 0.80 is approximately $\sqrt{\frac{0.8(1-0.8)}{10000}} = 0.004$.

Since we parametrize the trial in terms of alpha allocations whose levels sum to $\alpha = 0.025$, all of our proposed designs are asymptotically guaranteed to control the familywise Type I error as proved in (Rosenblum et al., 2015; Maurer and Bretz, 2013), and it is not necessary to penalize for violations of the required familywise Type I error rate in the manner of (Wason and Jaki, 2012; Wason et al., 2012).

4.4.3 Search using simulated annealing

We search for a minimizer $\mathscr{D}$ of $J(\mathscr{D})$ using Simulated Annealing (SA). The general form of SA is as follows. Given a trial design $\mathscr{D}$ as a reference point,

SA randomly perturbs $\mathscr{D}$ in order to generate a new proposal design $\mathscr{D}'$. If $J(\mathscr{D}') < J(\mathscr{D})$ then the proposal is "accepted," and $\mathscr{D}$ becomes the new reference point. If $J(\mathscr{D}') > J(\mathscr{D})$, then $\mathscr{D}'$ is accepted according to a certain probability, and discarded otherwise. The nonzero probability of exploring undesirable regions of the parameter space allows SA to avoid becoming stuck at local minima. As the algorithm progresses, new proposal designs $\mathscr{D}'$ are taken from a closer neighborhood around the reference design, and the probability of accepting inferior designs decreases. Both of these changes are modulated by a parameter known as the "temperature," which decreases with each iteration. We use the variant of SA implemented in the `optim` function in `R`, which is based on the algorithm of (Bélisle, 1992). We implemented SA in parallel across 100 nodes, each starting with a different random seed. Our implementation is "embarrassingly parallel" in that each node runs the SA algorithm independent of the others (i.e., without communication between nodes); when the SA search terminates for all nodes, we select the best design found.

The search space for $\mathscr{D}$ consists of a positive integer K; non-negative sample sizes n_k; alpha allocation proportions $\alpha_j^{(k)}$ summing to α; transition weights g_{ij} satisfying $\sum_{j \neq i} g_{ij} = 1$; and futility boundaries $f_j^{(k)}$. Separate searches are performed for the two hypothesis testing frameworks $\mathscr{H}^{MB}$ and $\mathscr{H}^{COV}$. One difficulty is that the dimension of this search space changes with the value of K, since greater values of K require additional sample sizes, efficacy boundaries, and futility boundaries. We give details on our method to address this issue in the supplemental materials.

The SA algorithm allows proposed design parameters to take any real values, which may violate the constraints on our search space of feasible designs. In

particular, since the alpha allocated to each test at each stage must be bounded between 0 and α, we instead use SA to search for the logit transform of the alpha allocated, i.e., $\log\{\alpha_j^{(k)}/(1-\alpha_j^{(k)})\}$. We then transform proposed values for $\text{logit}(\alpha_j^{(k)})$ back to the (0,1) interval, and rescale them to sum to α. In the same way, we search over the logit of the transition weights g_{ij}. Because we have only three hypotheses, alpha from one hypothesis can be reallocated to at most two other hypotheses, and so it is sufficient to simply search for g_{12}, g_{2C} and g_{C1}. These will uniquely determine the remaining transition weights according to $\sum_{j\neq i} g_{ij} = 1$. Non-negative and integer constraints for n_k and K are achieved by truncating and rounding respectively. Additionally, rather than searching for each individual n_k, we search across the space for N and separately search over the proportion of N allocated to each stage. Under this parametrization, the total sample size can naturally be changed without affecting the efficacy boundaries, as the efficacy boundaries depend only on the relative sample size at each stage. We refer to the resulting trial design as $\mathscr{D}_{SA}$.

Penalized approaches such as (Wason and Jaki, 2012; Wason et al., 2012), or approaches based on (4.3), will not necessarily guarantee that the resulting optimized design meets the power constraints in Section 4.4.1, as there may be cases where a small penalty is outweighed by a larger reduction in expected sample size. For the designs proposed by (Wason and Jaki, 2012; Wason et al., 2012), these concerns also apply to Type I error control.

To address the above issue, we built in an extra step to correct for cases where, after the SA algorithm completes, the resulting design $\mathscr{D}_{SA}$ fails to satisfy one or more of the power constraints. This step involves starting with $\mathscr{D}_{SA}$, and increasing only the total sample size parameter N. A binary search over N is

conducted to find the smallest value such that the constraints in Section 4.4.1 are met. During this search, all other elements of $\mathscr{D}_{SA}$ are held constant. When implementing the SA procedure in parallel, we apply this extra step after SA completes in each node. These supplemental searches also reduce the danger of choosing the tuning parameter λ in (4.3) to be too small. Our specific use of binary search is motivated by our empirical experience of $E_\Lambda(\tilde{n}(\mathscr{D}))$ being monotonically increasing in N for a variety of tested scenarios, and by the fact that the power constraints can always be satisfied by a sufficient increase to N.

In order to derive designs that are simpler to interpret and perform approximately optimally, we propose a two-step procedure for discovering efficacy and futility boundaries. First, we optimize as above. Next, we consider a lower dimensional parametrization that has a simpler form (e.g., with efficacy and futility boundaries restricted to vary smoothly rather than being allowed to oscillate wildly), and solve the same optimization problem in this restricted space. If the value of the objective function is very close to that attained in the unrestricted case, we report the simpler "structured" solution along with the "unstructured" one, as the former may be easier to communicate. We discuss our specific choice of structured boundaries in Section 4.5.

4.5 Results

We compare the performance of optimized adaptive designs against that of several more traditional designs. We find that optimized designs can offer substantial benefits, but that these benefits can be highly contingent on the delay time to the measurement of outcomes. Specifically, in the case the ADNI dataset,

the long delay time results in 1000 participants needing to be enrolled before the first participant outcome is measured. Less information is then available at interim decision points, and efficiency gains from any adaptive design are meager. In contrast, expected sample sizes can be substantially reduced in the MISTIE example, where the time to measurement is much faster relative to the trial enrollment rate. Here also, optimized adaptive designs offer a much greater benefit than more traditional adaptive designs.

We compare optimized adaptive designs against three types of traditional designs, which we denote as "standard one-stage designs," "optimized one-stage designs," and "standard multistage designs." We define standard one-stage designs as trials with equal alpha allocation and reallocation. We define optimized one-stage designs as trials where the alpha allocations and reallocations are found either through a grid search (for $\mathscr{H}^{COV}$) or through SA (for $\mathscr{H}^{MB}$). Finally, we define standard multistage designs as 5-stage trials with equal participant recruitment across stages, equal alpha allocation and reallocation across hypotheses, and futility boundaries set equal to zero. For standard multistage designs, the initial alpha allocations across stages were set according to the structured alpha spending function in Section 4.3.2, with ρ_{ej} set equal to either 1 or 3 for all j. These settings for ρ_{ej} result in boundaries similar to Pocock boundaries (Pocock, 1977) or O'Brien Fleming boundaries (O'Brien and Fleming, 1979) respectively (Jennison and Turnbull, 1999). For all comparison designs, the maximum sample size was selected to be the smallest value that satisfied the power constraints in Section 4.4.1.

4.5.1 MISTIE example

We first explore results for the MISTIE data example. We refer to those with small IVH as subpopulation 1, and those with large IVH are subpopulation 2. Based on prior research by Hanley (2012), the proportion of participants with small IVH (π_1) was projected to be 0.33. As many as 420 participants could be enrolled per year, from the combined population. The probability of a positive outcome under control was projected to be 0.290. Investigators aimed to satisfy the power constraints listed in Section 4.3 for $\delta^{\min} = 0.122$. Based on this, we set the variance of the outcome under control at $\sigma_{c1}^2 = \sigma_{c2}^2 = 0.290(1 - 0.290)$, and the variance of the outcome under treatment at $\sigma_{t2}^2 = \sigma_{t1}^2 = 0.412(1 - 0.412)$.

The first row of Figure 4.1 shows the initial (i.e., before any alpha-reallocation has taken place) z-statistic boundaries and per-stage sample sizes for the adaptive designs optimized for $\mathscr{H}^{COV}$ and for $\mathscr{H}^{MB}$. These boundaries are the result of the (unstructured) search for each individual alpha allocation and futility boundary. Initial efficacy boundaries for $\mathscr{H}^{COV}$ and $\mathscr{H}^{MB}$ are highly similar, each roughly resembling Pocock boundaries. Futility boundaries are similar across hypothesis testing frameworks as well, with futility boundaries for H_1 or H_2 being highest at the midpoint of the trial, and futility boundaries for H_C remaining low throughout the trial. Within a given design, symmetry between the futility boundaries for H_1 and H_2 is not necessarily to be expected, as $\pi_1 \neq \pi_2$.

We implemented the 2-step procedure described in the last paragraph of Section 4.4.3. Step 1 is the above optimization. Based on these results, we proposed the following structured form for the futility boundaries in step 2: let

$f_j^{(k)}$ be equal to $c_j + l_j \times (b_k - b_{k-1})$, where $b_k = \left(\frac{\sum_{k'}^{k} n_{k'}}{N}\right)^{\rho_{fj}}$, $b_0 = 0$, and l_j, c_j, and ρ_{fj} are design parameters for $j = 1, 2, C$. This form mirrors that of the parametric form for alpha allocation (see Section 4.3.2), but with additional shift parameters to capture the behavior discovered in the first row of Figure 4.1. We reapplied our SA procedure optimizing over the parameters c_j, l_j, and ρ_{fj} for each j, as well as the parameters w_j and ρ_{ej} as described in Section 4.3.2. The design discovered in this second stage had an expected sample size within 1.5% of the unstructured optimized design, and actually led to small improvements in expected sample size. The z-statistic boundaries and per-stage sample sizes for these optimized structured designs are shown in the second row of Figure 4.1.

Figure 4.2 shows the sample size distributions in the MISTIE data example for the optimized structured multistage designs, and for the comparison designs described above. The sample size distributions for multistage designs are shown as violin plots, and the fixed sample sizes of optimized and standard one-stage designs are shown as horizontal lines. All sample size distributions are calculated based on the prior distribution for the treatment effects in Section 4.4.1. Standard trials with Pocock boundaries improve expected sample size relative to a standard one-stage design (from 1891 to 1554 for $\mathscr{H}^{COV}$, and from 1885 to 1532 for $\mathscr{H}^{MB}$), but did not improve on optimized one-stage designs (1447 for $\mathscr{H}^{COV}$; 1443 for $\mathscr{H}^{MB}$). Expected sample sizes for standard trials with O'Brien Fleming boundaries were similar (1649 for $\mathscr{H}^{COV}$; 1670 for $\mathscr{H}^{MB}$), outperforming standard one-stage trials but not optimized one-stage trials. Relative to optimized one-stage designs, these two standard multi-stage

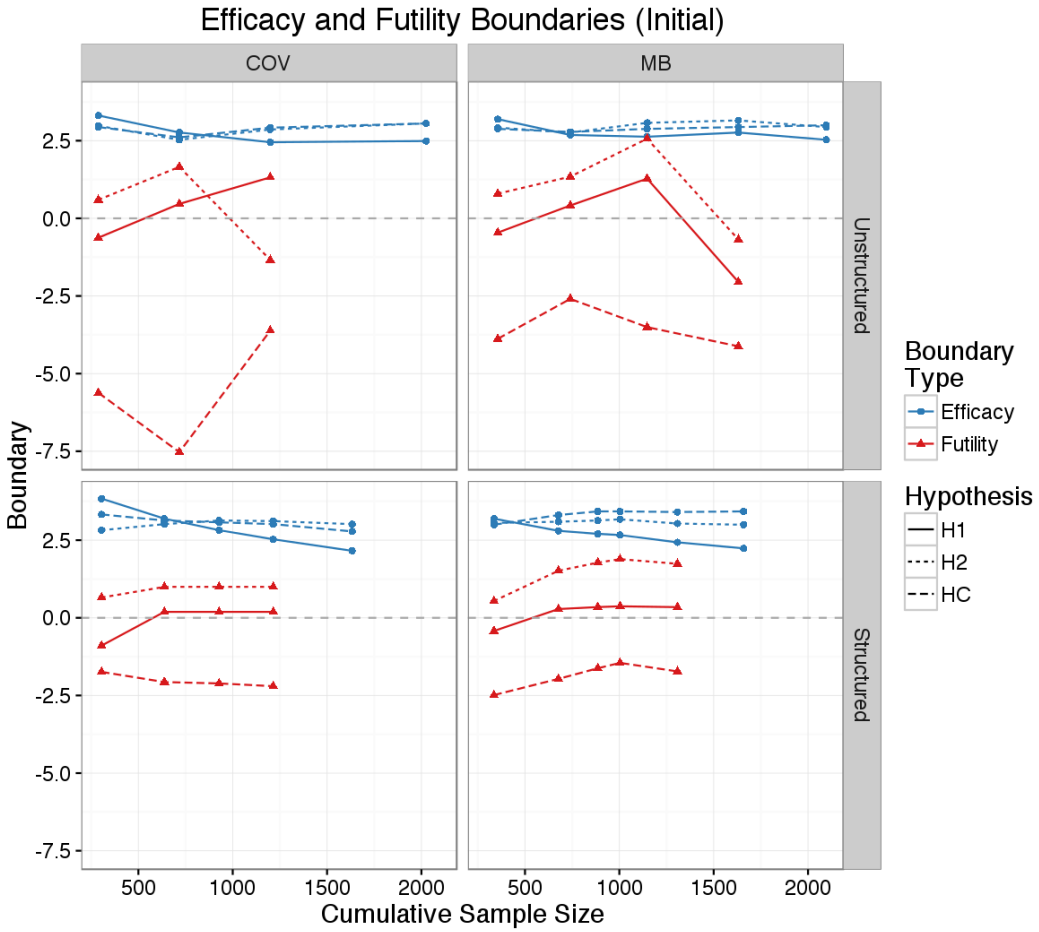

Figure 4.1: Design Parameters for the MISTIE Scenario - Here we show the initial efficacy and futility boundaries for the z-statistics, as well as per-stage sample sizes, for four different optimized trial designs (one in each panel). Dots and triangles mark the points at which interim analyses are scheduled to take place, with corresponding sample sizes on the x-axis. Each column of panels corresponds to a different hypothesis testing framework, with $\mathscr{H}^{COV}$ on the left and $\mathscr{H}^{MB}$ on the right. The top row of panels shows results from optimizing each boundary individually, while the second row shows the results from optimizing over a specific structured form for the boundaries. For $\mathscr{H}^{MB}$, the boundaries shown represent initial boundaries before any alpha reallocation. The alpha reallocation rules from the optimized designs are given in the supplemental materials, along with tables of the initial alpha allocations for all four designs.

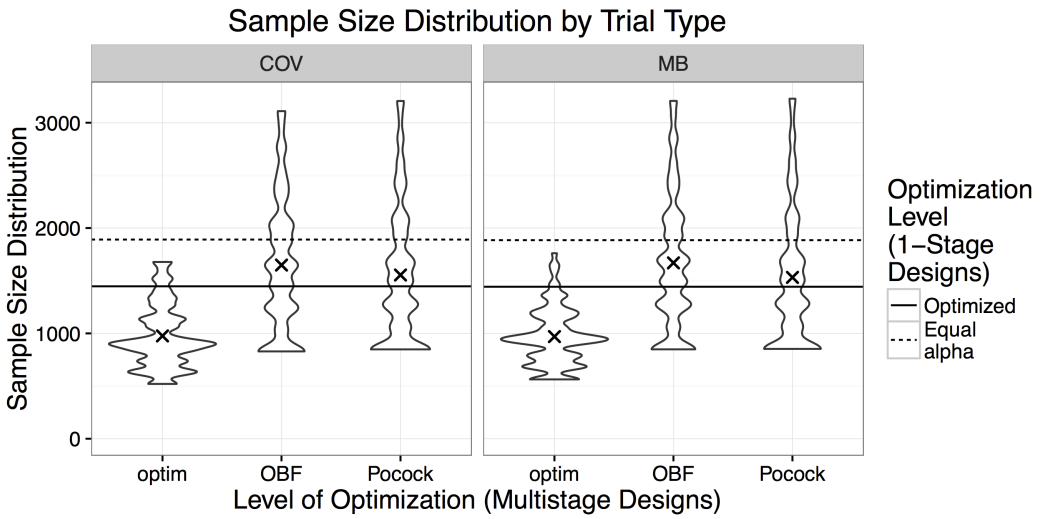

Figure 4.2: Trial Sample Sizes for the MISTIE Scenario - Violin plots are used to represent the sample size distributions for three types of multistage designs: optimized designs with structured boundaries (optim), O'Brien Fleming Boundaries (OBF), and Pocock boundaries (Pocock). These violin shapes represent smoothed histograms of the distribution of simulated sample sizes, aligned vertically for easier comparison with reference points. The sample size distribution is taken with respect to the prior for the treatment effects described in Section 4.4.1, with the mean sample size for each design shown as an "×" mark. As reference points, horizontal lines show the deterministic sample sizes from two types of one-stage designs (either with equal alpha allocation and reallocation, or with optimized alpha allocation and reallocation). Each panel corresponds to a different hypothesis testing framework, with $\mathscr{H}^{COV}$ on the left and $\mathscr{H}^{MB}$ on the right.

designs also come with the cost of a much higher maximum sample size. In contrast, adaptive designs optimized for $\mathscr{H}^{COV}$ and $\mathscr{H}^{MB}$ respectively reduced the expected sample size to 976 and to 970, with smaller increases to the maximum sample sizes (to 1678 and 1761 respectively).

Figure 4.3 shows approximate improvements to trial design performance in the MISTIE example at each iteration of the parallel, unstructured SA search. For each scenario, optimizations were parallelized across 100 computing nodes.

Each node was set to run for 5000 iterations or 24 hours, whichever occurred first. The curves in Figure 4.3 represent the trajectory of the cumulative minimum objective function value found by each parallel node. Quartiles with respect to the distribution of final objective function values are shown as horizontal lines. The figure is approximate in that no binary search corrections have yet been made to guarantee that power constraints are met. (See Section 4.4.3.) The most notable increases in performance occur in the early stages of SA, after which the distribution of performance across nodes remains relatively constant. This implies that a reduced number of search iterations might achieve similar performance if the temperature parameter of the search was set to decrease more slowly.

The quartile lines in Figure 4.3 can be used to estimate performance in cases where fewer computing resources would be available. For instance, if only 5 parallel nodes had been available, the probability of achieving a result below the first quartile would be approximately $(1 - 0.75^5) = 76\%$. Thus, at least some amount of parallelization appears to be an important component of the search.

4.5.2 ADNI example

We next consider simulations based on the ADNI data example. Here we denote non-carriers of the APOE $\epsilon 4$ allele as subpopulation 1, and participants who carry at least one allele as subpopulation 2. We set additional parameters based the subset of the ADNI data with baseline CDR-SB ≥ 0.5, of which 46.9% carry at least one APOE $\epsilon 4$ allele. We set the sample variance in the 2-year change in CDR-SB to be 3.44 for non-carriers ($\sigma_{c1}^2 = \sigma_{t1}^2 = 3.44$), and 3.72

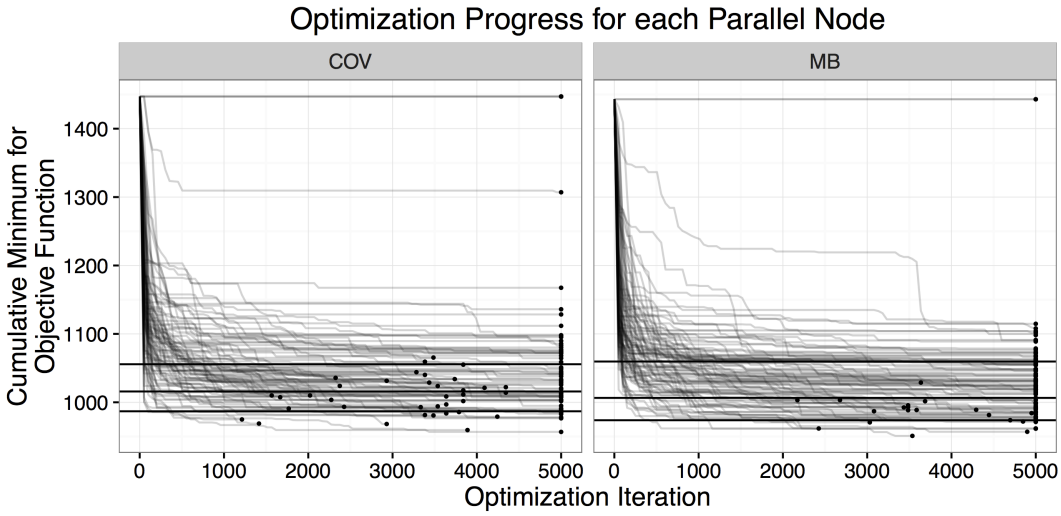

Figure 4.3: Objective Function over Parallel Search iterations, for the MISTIE Scenario - Each decreasing curve shows the trajectory of the cumulative minimum value of the objective function discovered by a parallel computing node. Black dots show the terminal of each node's trajectory. For nodes that did not complete 5000 iterations of SA within 24 hours, these dots mark the last iteration completed. Horizontal lines show the 0.25, 0.5, and 0.75 quantiles, respectively, for the final distribution of objective function values across the 100 parallel nodes. Each panel corresponds to a different hypothesis testing framework, with $\mathscr{H}^{COV}$ on the left and $\mathscr{H}^{MB}$ on the right.

for carriers ($\sigma_{c2}^2 = \sigma_{t2}^2 = 3.72$). The average change CDR-SB in the combined population was estimated at 1.41. The minimum clinical difference was set at a 30% reduction in this CDR-SB change, or $\delta^{\min} = 1.41 \times 0.3 = 0.42$. Based on our choice of outcome, the delay time from enrollment to outcome measurement (d) is exactly 2 years.

In contrast to the MISTIE example, no adaptive trial in the ADNI example was shown to lower expected sample size by more than 2% relative to an optimized one-stage design. As mentioned above, this can be largely attributed to the high enrollment required before any outcomes are measured. However, slight efficiency gains can still be made in terms of the trial's expected duration, as trials showing no treatment benefits can be stopped before waiting for all participant outcomes to be measured. Figure 4.4 shows performance comparisons for the ADNI example analogous to Figure 4.2, but with the y-axis showing the distribution of trial durations rather than the distribution of sample sizes. Here, multistage designs optimized for shorter trial durations were able to reduce expected duration by 8% using $\mathscr{H}^{MB}$, or 7% using $\mathscr{H}^{COV}$, relative to an optimized one-stage trial.

4.5.3 Alternative optimization algorithms

We also compared the performance of SA against other optimization algorithms available in the `optim` function in `R`. For each combination of testing procedure ($\mathscr{H}^{MB}$ or $\mathscr{H}^{COV}$), application (ADNI or MISTIE) and boundary form (structured or unstructured), each optimization method was allowed to run on 250 parallel nodes for either 4 hours or 2500 iterations, whichever occurred first. Rather than searching for the optimal number of stages (K), we fixed K within

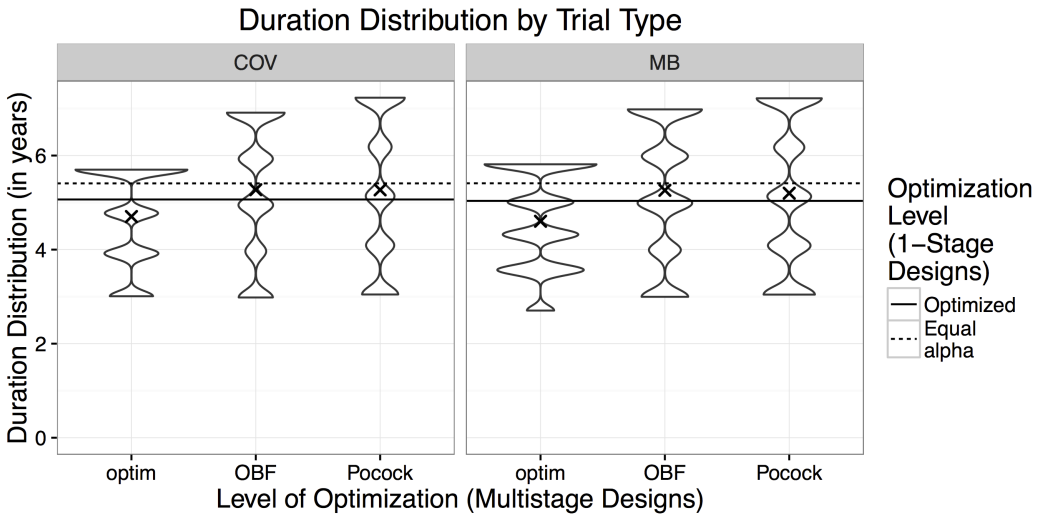

Figure 4.4: Trial Durations for the ADNI Scenario - Violin plots show the sample size distributions for three types of multistage designs: optimized designs with structured boundaries (optim), O'Brien Fleming Boundaries (OBF), and Pocock boundaries (Pocock). The duration distribution is taken with respect to the prior for the treatment effects described in Section 4.4.1, with the mean duration for each design shown as an "×" mark. As reference points, horizontal lines show the deterministic duration from two types of one-stage designs (either with equal alpha allocation and reallocation, or with optimized alpha allocation and reallocation). Each panel corresponds to a different hypothesis testing framework, with $\mathscr{H}^{COV}$ on the left and $\mathscr{H}^{MB}$ on the right.

a node at either 2, 3, 4, 5, or 6. These values for K were evenly distributed such that each unique configuration was allotted $250/5 = 50$ parallel nodes. The minimum objective function value across all 250 parallel nodes was recorded for comparison.

SA outperformed gradient methods such as BFGS, L-BFGS-B, and Conjugate Gradient by 5-8% in the ADNI example and 11-36% in the MISTIE example. Nelder-Mead and SA performed much more similarly. In the ADNI example, Nelder-Mead outperformed SA by approximately 2%. In the MISTIE example however, where more efficiency gains were available, SA outperformed Nelder-Mead by approximately 2%.

We also compared against a version of SA where the objective function for the current design $\mathscr{D}$ is re-evaluated at each comparison to a new candidate design $\mathscr{D}'$, as discussed in the conclusion of Branke et al. (2008). Such an approach will double the number of simulations required, but will decrease the probability that the algorithm becomes stuck at an inferior design where performance is initially over-estimated due to Monte Carlo error. This altered SA algorithm improved over gradient based methods, but was outperformed by both Nelder-Mead and by standard SA.

4.6 Discussion

We show empirical evidence that SA can yield adaptive enrichment trial designs with substantially lower expected sample sizes than a one-stage trial, or standard multistage designs with approximate Pocock or O'Brien Fleming boundaries. Relative to one-stage designs, optimized designs discovered here come at

the cost of smaller increases in maximum sample size. Much of the efficiency gain from optimization appears to be driven by changes to the futility boundaries. We use SA to compare approximate best-case implementations of covariance-based and alpha-reallocation-based trial designs, and find such best-case designs to be similar in both their design parameters and their performance.

One exciting area of future work is to more actively account for the Monte Carlo simulation error in our objective function evaluations. Some optimization methods leverage noise present in the objective function, or add noise to the objective function (Kushner, 1987; Maryak and Chin, 2001), in order to increase the probability of reaching a global minimum. In the specific context of SA, (Fink, 1998; Branke et al., 2008) argue that noise in the objective function is analogous to having a higher temperature parameter.

4.7 Acknowledgements

This chapter is joint work with Michael Rosenblum, and was supported by the Participant-Centered Outcomes Research Institute (ME-1306-03198) and the U.S. Food and Drug Administration (HHSF223201400113C). We thank Daniel Hanley and Michela Gallagher for providing summary statistics from the MISTIE and ADNI data sets, respectively.

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National

Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

This work is solely the responsibility of the authors and does not represent the views of the above people and agencies.

Chapter 5

Supplement to Chapter 3

5.1 Block matrix algebra for when the data cannot fit into memory

When the number of measurements is especially large (i.e., p >10,000), it is common that either the $p \times n$ data matrix $\mathbf{Y}$, or the $p \times B$ matrices storing each of the K fitted PCs across bootstrap samples, may be too large to store in working memory. This issue can be remedied by using block matrix algebra to subdivide the SVD computation into a series of low memory steps.

The standard algorithm for calculating the SVD of a high dimensional $p \times n$ matrix $\mathbf{Y}$ begins by first calculating the $n \times n$ matrix $\mathbf{Y}'\mathbf{Y}$. Calculating the SVD of $\mathbf{Y}'\mathbf{Y}$ yields $\mathbf{U}\mathbf{D}^2\mathbf{U}'$. The matrix $\mathbf{V}$ can then be calculated as $\mathbf{V} = \mathbf{Y}\mathbf{U}\mathbf{D}^{-1}$. When p is much larger than n, the computational complexity of this method is $O(pn^2)$.

For the case where $\mathbf{Y}$ is too large to be stored in working memory, define $\{s_1, s_2, ...s_m\}$ to be the set of m indexing vectors, each of length p/m, such that the p-length concatenated vector $(s_1, s_2, ...s_m)$ is equal to vector $(1, 2, 3, ...p)$. Note that $\mathbf{Y}$ can now be partitioned as $\mathbf{Y}' = [\mathbf{Y}'_{[s_1,]}\ \mathbf{Y}'_{[s_2,]}\ ...\ \mathbf{Y}'_{[s_m,]}]$. The matrix

$\mathbf{Y}'\mathbf{Y}$ can be calculated as $\sum_{i=1}^{m} \mathbf{Y}'_{[s_i,]}\mathbf{Y}_{[s_i,]}$, where each term of the sum can be calculated separately. We can similarly partition $\mathbf{V}' = [\mathbf{V}'_{[s_1,]}\ \mathbf{V}'_{[s_2,]}\ ...\ \mathbf{V}'_{[s_m,]}]$, with $\mathbf{V}_{[s_m,]} = \mathbf{Y}_{[s_m,]}\mathbf{U}\mathbf{D}^{-1}$. Neither the entire matrix $\mathbf{Y}$ nor the entire matrix $\mathbf{V}$ need ever be stored in memory. (Zipunnikov et al., 2011a)

For the bootstrap percentile intervals described in section 3.3.1 of the main paper, note that the bootstrap CIs for each block of $\mathbf{V}$ can be calculated separately. In each bootstrap sample, the fitted PCs can be partitioned as

$$\mathbf{V}^{b\prime}_{[,1:K]} = [\mathbf{V}^{b\prime}_{[s_1,1:K]}\ \mathbf{V}^{b\prime}_{[s_2,1:K]}\ \cdots\ \mathbf{V}^{b\prime}_{[s_m,1:K]}]$$

and calculated according to the relation $\mathbf{V}^{b}_{[s_i,1:K]} = \mathbf{V}_{[s_i,]}A^{b}_{[,1:K]}$. The bootstrap percentiles for the elements of each partition of the PC matrix can be calculated separately, without storing the high dimensional the bootstrap distribution of $\mathbf{V}^{b}_{[,1:K]}$ in working memory. Of the different CIs and CRs proposed in section 3.3 of the main paper, percentile intervals form the only case where memory constraints become a potential issue in the bootstrap calculations. For the moment-based pointwise intervals, as well as the other confidence regions discussed in section 3.3, only the low dimensional bootstrap distribution of $A^{b}_{[,1:K]}$ is required.

5.2 Random preconditioning for when the SVD fails to converge

We find in approximately 4% of bootstrap samples from the MRI dataset, that although a solution to the SVD of $\mathbf{D}\mathbf{U}'\mathbf{P}^b$ exists, the SVD function fails to converge. We handle these cases by randomly preconditioning the matrix $\mathbf{D}\mathbf{U}'\mathbf{P}^b$,

and reapplying the SVD function. We then adjust the output of this preconditioned SVD operation to find the solution for the SVD of the original matrix $\mathbf{DU}'\mathbf{P}^b$.

The specific steps of this adjusted SVD algorithm are described below, in terms of their application to an arbitrary $n \times m$ matrix $\mathbf{\Sigma}$.

- Step 1: To find $svd(\mathbf{\Sigma})$, first generate a two random orthonormal matrices $\mathbf{Q}_n$ and $\mathbf{Q}_m$, of dimension $n \times n$ and $m \times m$ respectively. Each matrix can be obtained by taking the QR decomposition of a square matrix of random normal noise.

- Step 2: Calculate the SVD of $\mathbf{Q}_n'\mathbf{\Sigma}\mathbf{Q}_m$, and denote the result as $\mathbf{VDU}'$.

- Step 3: If this SVD operation also fails to converge, repeat steps 1-2 until either a solution is found, or a pre-specified maximum number of attempts is reached. We generally find that a single iteration is sufficient.

- Step 4: Write the SVD of $\mathbf{\Sigma}$ as $(\mathbf{Q}_n\mathbf{V})\mathbf{D}(\mathbf{Q}_m\mathbf{U})'$.

Note $(\mathbf{Q}_n\mathbf{V})$ and $(\mathbf{Q}_m\mathbf{U})$ are both orthonormal, $\mathbf{D}$ is diagonal, and

$$(\mathbf{Q}_n\mathbf{V})\mathbf{D}(\mathbf{Q}_m\mathbf{U})' = \mathbf{Q}_n\mathbf{VDU}'\mathbf{Q}_m' = \mathbf{Q}_n\mathbf{Q}_n'\mathbf{\Sigma}\mathbf{Q}_m\mathbf{Q}_m' = \mathbf{\Sigma}$$

So $(\mathbf{Q}_n\mathbf{V})\mathbf{D}(\mathbf{Q}_m\mathbf{U})'$ is indeed a solution to the SVD of $\mathbf{\Sigma}$. If the SVD of $\mathbf{\Sigma}$ is unique, then $(\mathbf{Q}_n\mathbf{V})\mathbf{D}(\mathbf{Q}_m\mathbf{U})'$ is the unique solution to the SVD.

When $\mathbf{\Sigma}$ is a square matrix, this procedure can be simplified by letting $\mathbf{Q}_m = \mathbf{Q}_n$. The procedure can also be made slightly faster by replacing $\mathbf{Q}_m$ and $\mathbf{Q}_n$ with random permutation matrices.

5.3 Centering bootstrap samples by centering scores

Centering the $p \times n$ matrix $\mathbf{Y}$ can be achieved by right multiplying by $(\mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n')$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix, and $\mathbf{1}_n$ is the n-length vector of ones. Since $\mathbf{Y}(\mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n') = \mathbf{VDU}'(\mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n')$, centering $\mathbf{Y}$ is equivalent to centering the $n \times n$ matrix of scores, $\mathbf{DU}'$.

Similarly, consider the bootstrap sample $\mathbf{Y}^b = \mathbf{YP}^b$. Because $\mathbf{YP}^b(\mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n') = \mathbf{VDU}'\mathbf{P}^b(\mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n')$, centering $\mathbf{Y}^b$ is equivalent to centering the $n \times n$ matrix of resampled scores, $\mathbf{DU}'\mathbf{P}^b$. Instead of taking the SVD of the resampled scores, we can simply take the SVD of the resampled *and centered* scores.

5.4 Changing the sign of bootstrap PCs - dot product v. correlation

We often wish to switch the sign of a bootstrap PC, $\mathbf{V}^b_{[,k]}$, to better align it with its corresponding sample PC, $\mathbf{V}_{[,k]}$. Switching the sign based on the cross product between $\mathbf{V}^b_{[,k]}$ and $\mathbf{V}_{[,k]}$ can yield a different decision than switching based on the correlation between $\mathbf{V}^b_{[,k]}$ and $\mathbf{V}_{[,k]}$. In this section, we compare cases where these two methods disagree, and argue that the results of the dot product approach are more interpretable.

The left panel of Figure 5.1 shows a case where $cor(\mathbf{V}_{[,k]}, \mathbf{V}^b_{[,k]}) < 0$, but $\mathbf{V}'_{[,k]}\mathbf{V}^b_{[,k]} > 0$. Here, the correlation rule would suggest that the sign of $\mathbf{V}^b_{[,k]}$ be inverted, but the dot product rule would imply that the sign should not be inverted. The right panel shows the opposite case, where $cor(\mathbf{V}_{[,k]}, \mathbf{V}^b_{[,k]}) > 0$,

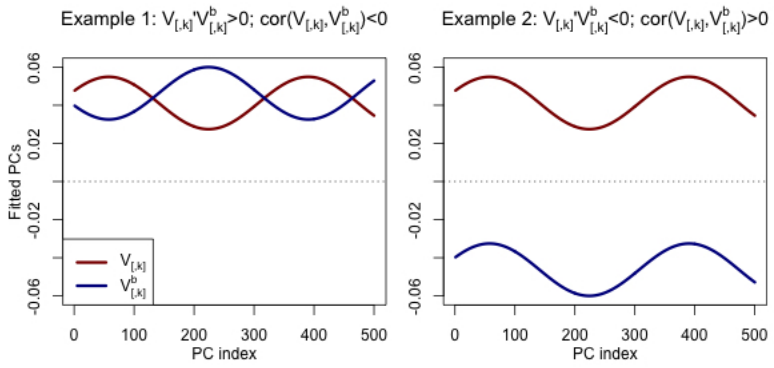

Figure 5.1: Axis reflections for $\mathbf{V}^b_{[,k]}$

implying that no sign inversion should be done, but $\mathbf{V}'_{[,k]}\mathbf{V}^b_{[,k]} < 0$, implying that an inversion should in fact be done. In both examples, we find the results of the dot product rule to be more intuitive. For PCs that are fairly flat, the correlation rule has the potential to create bimodal bootstrap distributions of $\mathbf{V}^b_{[,k]}$ on either side of the zero line.

5.5 Supplemental figures for the EEG and MRI Datasets

Figure 5.2 shows approximate reconstructions of the observed subjects' EEG measurements using only the K leading PCs, for $K = 1, 2, 5$, and 391. We see that the more PCs are included in the approximation, the more variability is retained from the original dataset. In the first panel ($K = 1$), the variability in the reconstructed dataset is roughly due to different within-subject average NP_δ levels. This panel also shows the average NP_δ across subjects, denoted by $\boldsymbol{\mu}$. In panel 2 we add in variability attributable to the second PC ($K = 2$), and see that reconstructed NP_δ measurements now also vary in terms of broad

oscillatory patterns in the early stages of sleep. Panel 3 shows variability due to the first five PCs ($K = 5$), and highlights how the primary patterns in NP_δ variability take place in the first three hours of the night. The final plot shows the full reconstruction of the dataset with all $n - 1$ PCs, and contains the most overall variability.

Figure 5.3 shows the cumulative variance explained by the first 30 PCs of the EEG dataset, and by the first 30 PCs of the MRI dataset. These curves are proportional to the cumulative sum of the eigenvalues of the sample covariance matrices for each dataset.

5.6 Additional simulation results

5.6.1 Pointwise interval coverage

Here we discuss the simulation results for pointwise confidence interval coverage rates in the baseline simulation scenario, with $p = 900$, $n = 392$, the empirical residual variance, and the empirical eigenvalue spacing. The line plots on the right of Figure 5.4 show coverage rates for the each of the p elements of the three PCs. Pointwise coverage for all elements of PC1 is very close to 95%. For both PC2 and PC3, the moment-based intervals consistently give close to 95% coverage, but the percentile intervals appear to give poor coverage in certain regions. This may be an artifact, however, due to how the percentile interval responds to skewness in the underlying bootstrap distribution. Adjusted percentile intervals, such as the BC_a interval (Efron, 1987), might account for this apparent coverage problem. It is possible that the difficulty in estimating coverage is also affected by the spacing of the eigenvalues – PC1 corresponds to

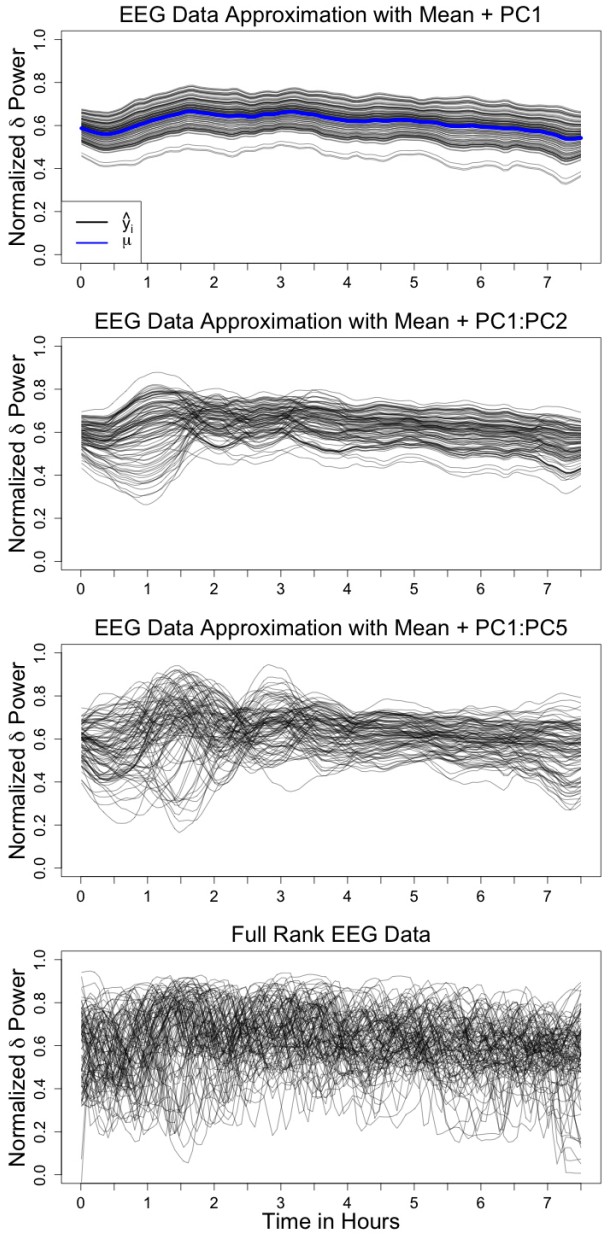

Figure 5.2: Reconstructions of EEG data with leading PCs - The first three panels respectively show approximations constructed using the first PC, the first two PCs, and the first five PCs. The first panel also shows the mean NP_δ across subjects, denoted by $\boldsymbol{\mu}$. The bottom panel uses all of the PCs to reconstruct the sample points exactly. To avoid over-plotting, reconstructions are shown only for a random subsample of 100 subjects.

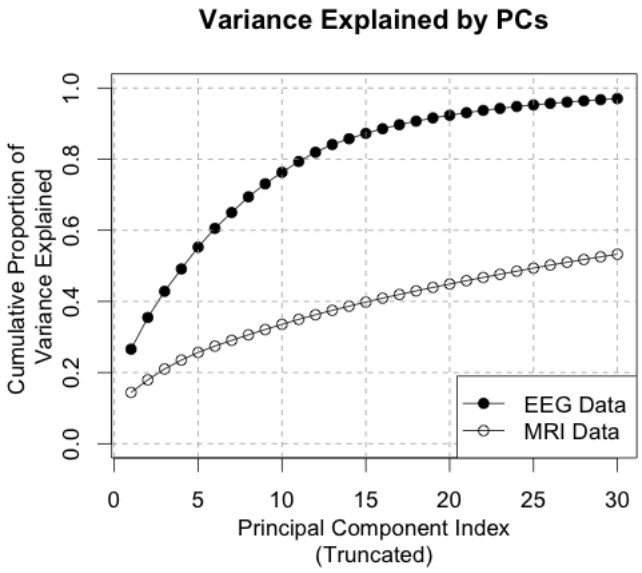

Figure 5.3: Cumulative proportion of variance explained by the first 30 PCs.

an eigenvalue that is clearly differentiated, while the eigenvalues for the second two components are less clearly differentiated from the remaining eigenvalues.

The violin plots on the left side of Figure 5.4 show the distribution of coverage rates across the PC curves as we vary the sample size and eigenvalue spacing. In this panel, the dimensionality (p) is fixed at 900, and only the empirical residual noise variance level (σ^2) is used, but results were very similar for alternate levels of dimensionality and residual variance. Coverage rates for all regions of the PC curves converges to 95% as sample size increases. The coverage is also more accurate when the eigenvalues are well spaced, such as when the first PC is being estimated, or when the parametric spacing for the eigenvalues is used.

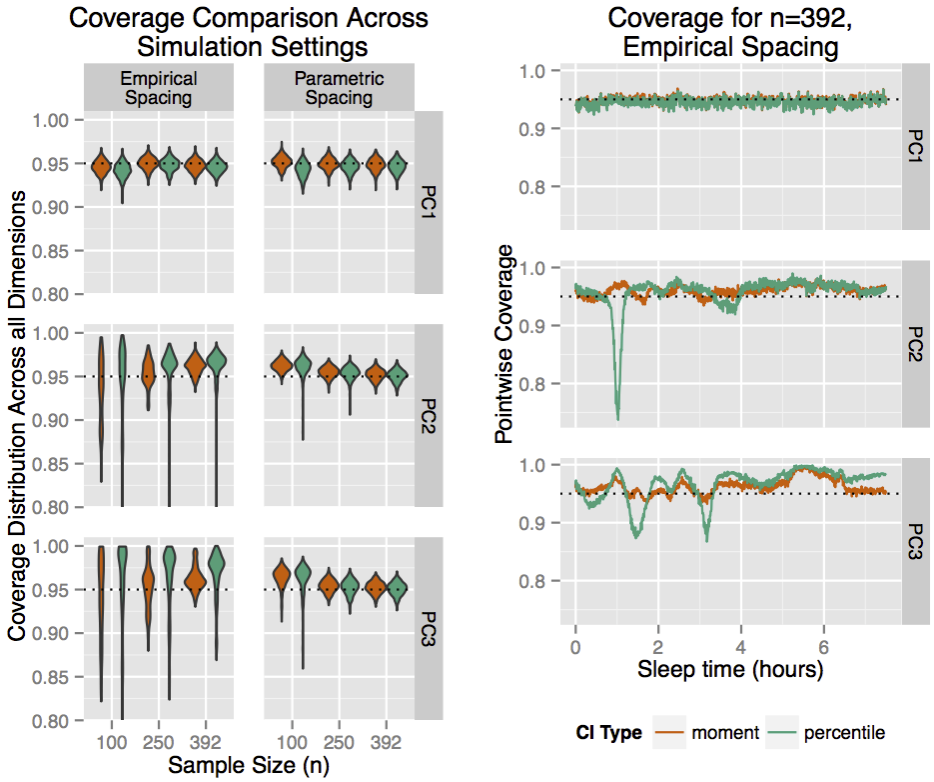

Figure 5.4: Pointwise coverage of the PCs - Pointwise bootstrap-based CIs can be calculated for each of the p dimensions of each PC. The violin plots on the left show the distribution of coverage rates across each of the p CIs, under different simulation settings (p fixed at 900). Simulation cases using the empirical eigenvalue spacing are shown on the left column of violin plots, and simulation cases where where each PC explains half as much variance as the previous PC are shown on the right column. For ease of viewing, coverages are cropped at 80%. This resulted in 5.0%, 2.3% and 1.3% of coverage rates being cropped out for the PC2 percentile intervals, for $n = 100, 200$ and 300 respectively. The lowest simulated coverage rates in these respective cases were 52.1%, 66.9%, and 74.1%. For PC3, 4.6% of coverage rates were cropped from the figure for $n = 100$, with the minimum coverage rate occurring at 69.7%. The line plots on the right further explore coverage rates for the specific simulation setting of $n = 392$, $p = 900$, and the empirical eigenvalue spacing. Coverage rates are shown for each of the p CIs, with the x-axis corresponding to the p-dimensional PC element index (time). In both sets of plots, rows correspond to the PC being estimated.

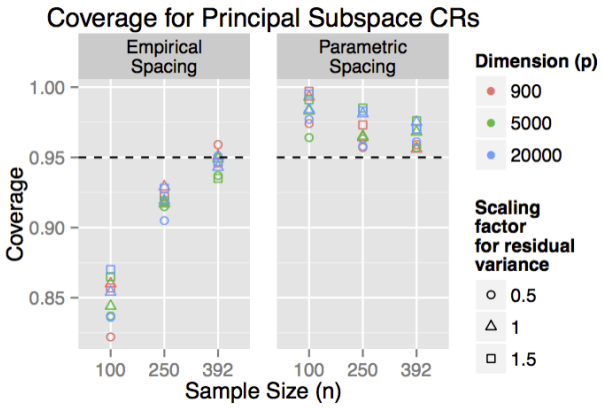

Figure 5.5: Coverage of CRs for the principal subspace

5.6.2 Coverage of confidence regions for principal subspace

Figure 5.5 shows coverage rates of confidence regions (CRs) for the principal subspace (see section 3.3.3 of the main paper). Coverage generally improves when the eigenvalues are well spaced and when sample size increases.

5.6.3 Coverage of best linear unbiased predictors for the scores

Our data was generated under the model $\mathbf{y} = \mathbf{\Psi s} + \epsilon$, where $\mathbf{y}$ is a p-dimensional outcome for a simulated subject, $\mathbf{s}$ is a K_0-dimensional vector of random, subject-specific scores, $\mathbf{\Psi}$ is the $(p \times K_0)$ matrix of true basis vectors, and ϵ is a p-dimensional vector of random errors. In calculating the best linear unbiased predictors (BLUPs) for $\mathbf{s}$, we assume $\mathbf{s} \sim N(0, \mathbf{G})$ and $\epsilon \sim N(0, \mathbf{I}_p\sigma^2)$, where $\mathbf{G}$ is diagonal with diagonal elements $(\lambda_1, \lambda_2, \ldots, \lambda_{K_0})$. Thus, $\mathbf{y}$ and $\mathbf{s}$ form a joint multivariate normal with

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{s} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{\Psi G \Psi'} + \mathbf{I}_p\sigma^2 & \mathbf{\Psi G} \\ \mathbf{G \Psi'} & \mathbf{G} \end{pmatrix} \right)$$

And the BLUP is given by the conditional expectation of y given s

$$E(\mathbf{s}|\mathbf{y}) = \mathbf{G}\mathbf{\Psi}'(\mathbf{\Psi}\mathbf{G}\mathbf{\Psi}' + \mathbf{I}_p\sigma^2)^{-1}y \tag{5.1}$$

By the Sherman–Morrison–Woodbury formula, the inverse in Equation (5.1) is equal to

$$\begin{aligned}(\mathbf{\Psi}\mathbf{G}\mathbf{\Psi}' + \mathbf{I}_p\sigma^2)^{-1} &= \mathbf{I}_p\sigma^{-2} - \mathbf{I}_p\sigma^{-2}\mathbf{\Psi}(\mathbf{G}^{-1} + \mathbf{\Psi}'\mathbf{I}_p\sigma^{-2}\mathbf{\Psi})^{-1}\mathbf{\Psi}'\mathbf{I}_p\sigma^{-2} \\ &= \mathbf{I}_p\sigma^{-2} - \mathbf{\Psi}(\mathbf{G}^{-1} + \mathbf{I}_{K_0}\sigma^{-2})^{-1}\mathbf{\Psi}'\sigma^{-4}\end{aligned}$$

Note that $(\mathbf{G}^{-1} + \mathbf{I}_{K_0}\sigma^{-2})^{-1}$ is diagonal with diagonal elements $((\lambda_1^{-1} + \sigma^{-2})^{-1}, (\lambda_2^{-1}+\sigma^{-2})^{-1}, \ldots, (\lambda_{K_0}^{-1}+\sigma^{-2})^{-1})$. Now, Equation (5.1) can be calculated as

$$\begin{aligned}E(\mathbf{s}|\mathbf{y}) &= \mathbf{G}\mathbf{\Psi}'(\mathbf{I}_p\sigma^{-2} - \mathbf{\Psi}(\mathbf{G}^{-1} + \mathbf{I}_{K_0}\sigma^{-2})^{-1}\mathbf{\Psi}'\sigma^{-4})y \\ &= \mathbf{G}(\mathbf{\Psi}'\sigma^{-2} - (\mathbf{G}^{-1} + \mathbf{I}_{K_0}\sigma^{-2})^{-1}\mathbf{\Psi}'\sigma^{-4})y \\ &= \mathbf{G}(\mathbf{I}_{K_0}\sigma^{-2} - (\mathbf{G}^{-1} + \mathbf{I}_{K_0}\sigma^{-2})^{-1}\sigma^{-4})\mathbf{\Psi}'y \end{aligned} \tag{5.2}$$

In each bootstrap sample, we estimate the BLUPs using the empirical BLUPs (EBLUPs). This estimator consists of plugging the empirical estimates of $\mathbf{G}$, $\mathbf{\Psi}$, and σ^2 into Equation (5.2) (Fitzmaurice et al., 2012). We use $(\mathbf{D}^b_{[k,k]})^2(1/(n-1))$ to estimate λ_k, $\mathbf{V}^{\mathbf{b}}_{[,1:K_0]}$ to estimate $\mathbf{\Psi}$, and $\sum_{K_0+1}^{n}(\mathbf{D}^b_{[k,k]})^2(1/(n-1))(1/p)$ to

estimate σ^2.[1]We then create percentile and moment-based CIs from the bootstrap distribution of the EBLUPS. Coverage of these CIs under different circumstances is shown in Figure 5.6.

5.6.4 Simulated accuracy of sample principal components

In each simulated sample, we recorded the angle between each sample PC and its corresponding population PC. We also record the angle between each sample PC and the subspace spanned by all K_0 population PCs. For each simulated scenario, Figure 5.7 shows the resulting 95% percentiles for these angles – in 95% of simulated samples, the angles were less than or equal to the ones shown here.

5.7 Computation times for bootstrap PCA

We tested the speed of our bootstrap PCA procedure for several combinations of sample size (n) and dimensionality (p). Varying n and p was achieved by using subsets of the measurements and subjects from the MRI dataset. All calculations were run on a standard laptop (2.5GHz Intel Core i5, 12 Gb memory), without parallelization.

Figure 5.8 shows the results of these tests. We compare our proposed methods against an approximate "brute force" calculation time, which is attained by multiplying the calculation time for the first 3 sample PCs by the number of bootstrap samples ($B = 1000$). This approximation is conservative in that it does not include time required for saving and loading the p-dimensional

[1]Note that $\mathbf{V^b}_{[,1:K_0]}$ only appears in the form $\mathbf{V^{b'}}_{[,1:K_0]}y = \mathbf{A^{b'}}_{[,1:K_0]}\mathbf{V'y}$, where $\mathbf{V'y}$ can be precalculated before the bootstrap procedure.

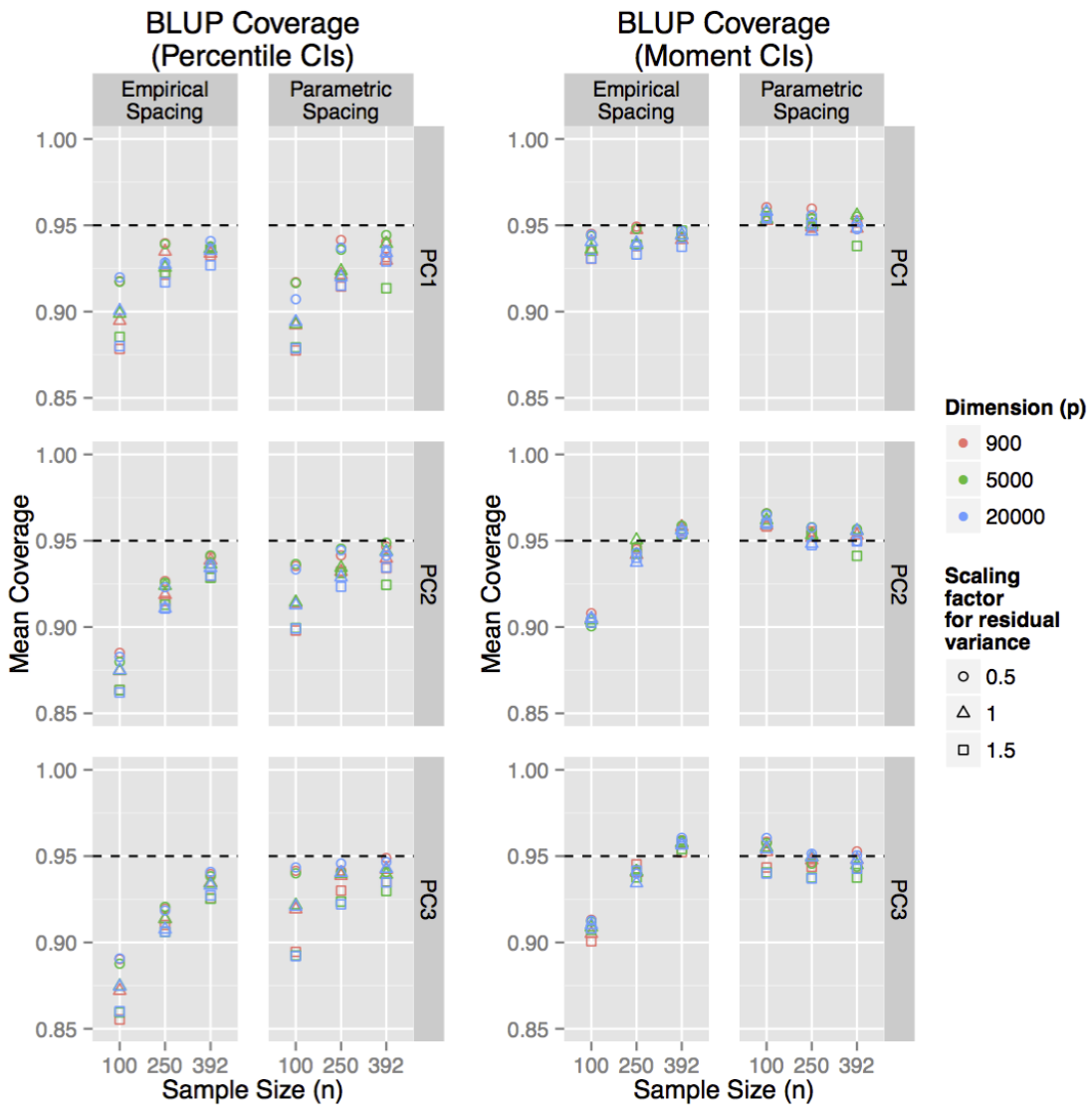

Figure 5.6: Coverage for Best Linear Unbiased Predictors (BLUPs) - For a given simulation scenario, the y-axis shows the average coverage across all BLUPs from all simulations. Moment-based CIs are shown on the left, and percentile CIs are shown in the right.

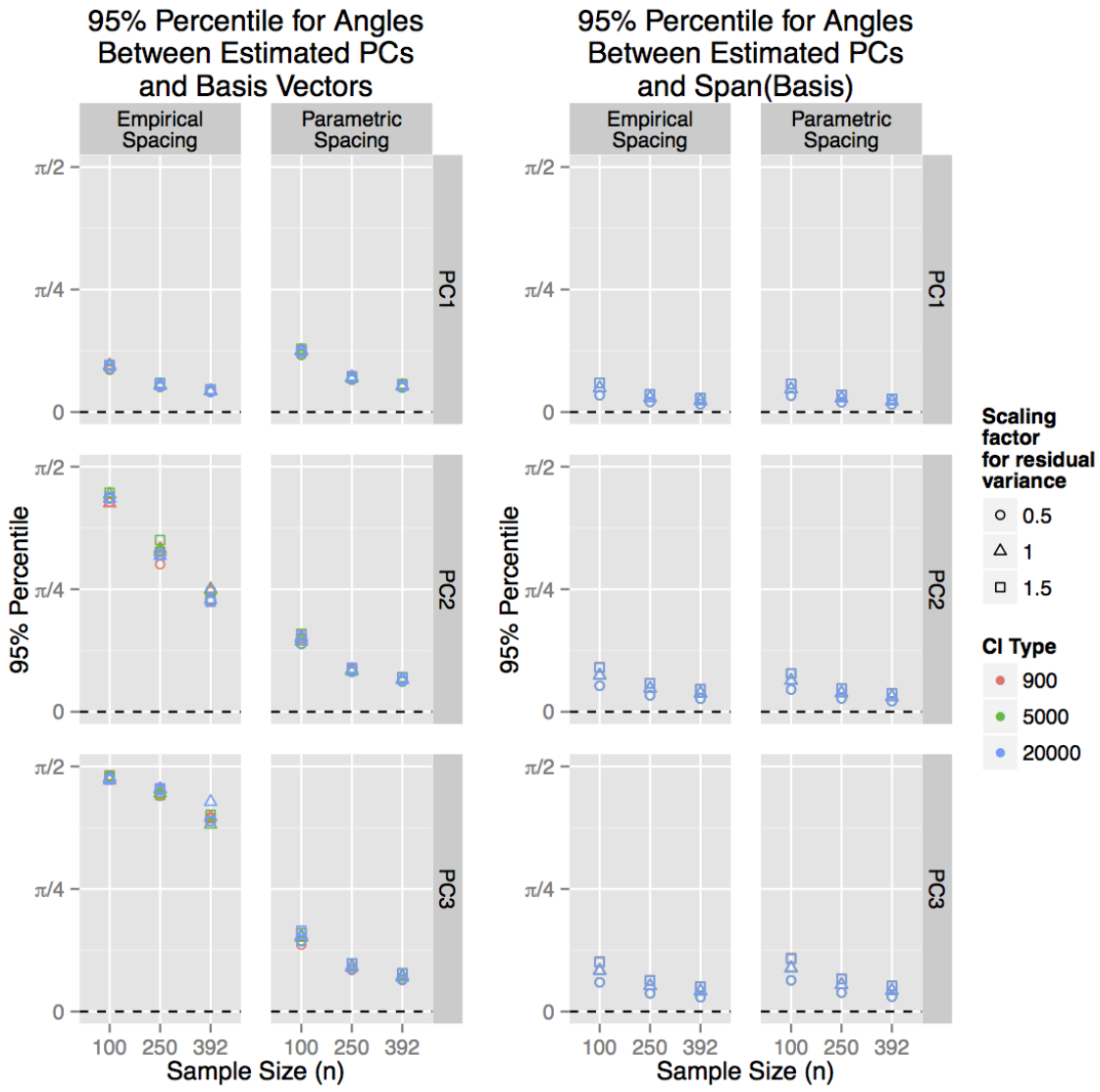

Figure 5.7: 95% Percentiles for angles between estimated PCs and generating basis

bootstrap PCs. Still, our methods offer significant speed improvements over the approximate brute force method in all tested scenarios. In particular, for the most computationally demanding scenario tested (p = 2,979,666; $n = 352$), pointwise percentile intervals based on the full bootstrap distribution of the PCs were calculated in 118 minutes using our method, as opposed to 5,693 minutes (3.95 days) with the brute force method. Calculating bootstrap standard errors with our method took only 47 minutes.

While the brute force method can be parallelized on a high powered computing cluster to reduce the total elapsed calculation time, the parallelization procedure will incur bottlenecks when multiple nodes attempt to simultaneously load the sample data files into memory. The sample data files will only be able to be accessed by one node at a time. This is an especially relevant problem for the high dimensional scenario, when the data must be stored as a set of block matrices that are loaded into memory sequentially (see section 5.1 of these supplemental materials). In contrast, our proposed method for fast, exact bootstrap PCA can be parallelized without incurring these bottlenecks, as each node only needs to import the $n \times n$ matrix of sample scores ($\mathbf{DU}'$).

5.8 Elliptical confidence regions on the hypersphere

One potential method for describing the dominant patterns in bootstrap PC variability, is to use p-dimensional elliptical confidence regions (CRs) of the form

$$\{\mathbf{x} \in S_p : (\mathbf{x}-\mathbf{V}_{[,k]})' Cov(\mathbf{V}^b_{[,k]})^- (\mathbf{x}-\mathbf{V}_{[,k]}) \leq q((\mathbf{V}^b_{[,k]}-\mathbf{V}_{[,k]})' Cov(\mathbf{V}^b_{[,k]})^- (\mathbf{V}^b_{[,k]}-\mathbf{V}_{[,k]}), \alpha)\}$$

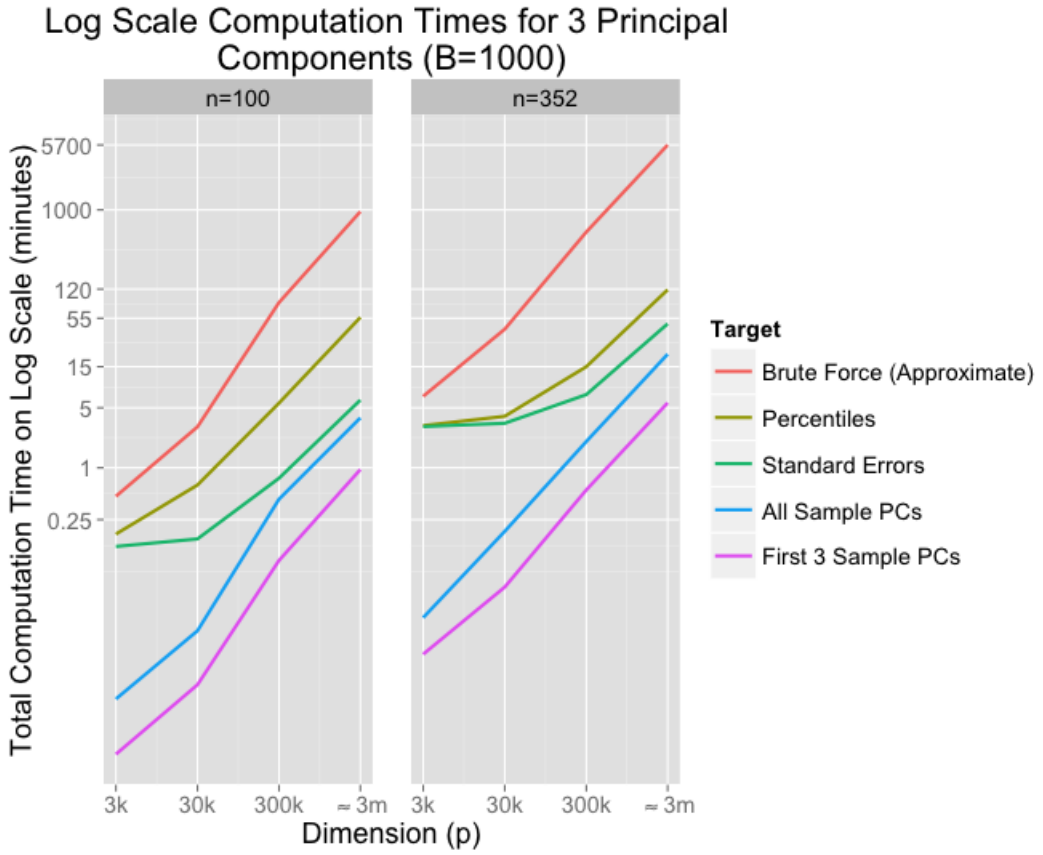

Figure 5.8: Computation times for bootstrap PCA - The two plots show computation times for sample sizes of 100 (left) and 352 (right). The horizontal axis shows the dimensionality (p = 3,000; 30,000; 300,000; and 2,979,666) and the vertical axis shows total elapsed computation time of each method. The spacing for both axes is on the log scale, in base 10. Computation times are shown for calculating the first 3 sample PCs, all n sample PCs, bootstrap standard errors, and bootstrap percentiles. For the bootstrap standard errors and percentiles, the computation time shown includes the time required for the full SVD of the original sample. An approximation of the time required to calculate the bootstrap distribution of the PCs using standard methods is also shown.

Where S_p is the p-dimensional hypersphere, $q(y^b, \alpha)$ is the $100\alpha^{th}$ percentile of the bootstrap variable y^b, $Cov(\mathbf{V}^b_{[,k]})$ is the $p \times p$ bootstrap covariance matrix of the k^{th} PC, and $Cov(\mathbf{V}^b_{[,k]})^-$ is the generalized inverse of $Cov(\mathbf{V}^b_{[,k]})$.

Note that the use of the generalized inverse, or some form of regularization, is required, as the covariance matrix $Cov(\mathbf{V}^b_{[,k]})$ is not full rank and not invertible. As a result, these CRs will not describe sampling variability in directions orthogonal to the span of the observed sample points. Note also that $Cov(\mathbf{V}^b_{[,k]})^- = (\mathbf{V}Cov(\mathbf{A}^b_{[,k]})\mathbf{V}')^- = \mathbf{V}(Cov(\mathbf{A}^b_{[,k]})^-)\mathbf{V}'$. Thus, the above CR is equivalent to the easily calculable region

$$\{\mathbf{x} \in S_p : (\mathbf{V}'\mathbf{x}-\delta_k)'Cov(\mathbf{A}^b_{[,k]})^-(\mathbf{V}'\mathbf{x}-\delta_k) \leq q((\mathbf{A}^b_{[,k]}-\delta_k)'Cov(\mathbf{A}^b_{[,k]})^-(\mathbf{A}^b_{[,k]}-\delta_k), \alpha)\}$$

Where δ_k is the k^{th} column of the $n \times n$ identity matrix. These elliptical CRs can be fully defined by the length and directions of their primary axes, which, in the case of spacial data, can be plotted on the p-dimensional scale.

Chapter 6

Supplement to Chapter 4

6.1 Changing dimension of search space

Because we have several stage-specific design parameters, the dimension of our search space for an optimal design depends on the number of stages K. Within our simulated annealing (SA) search, a newly proposed trial design $\mathscr{D}'$ may require an expansion or contraction of the search space, according to its proposed value for K. In order to address this, we restrict K to be less than or equal to 10, and maintain length-10 lists for the efficacy boundaries, futility boundaries, and per-stage sample sizes of each proposed design. In any one iteration of SA only the first K' elements of these lists are used, where K' is the proposed value for K at that iteration. For example, if the proposed design $\mathscr{D}'$ contained the length-10 list of stage-specific futility boundaries for H_C of (-10,-9,-8,-7,-6,-5,-4,-3,-2,-1), and the proposed value of 6 for K, then the stage-specific futility boundaries for H_C used in evaluating $J(\mathscr{D}')$ would be (-10,-9,-8,-7,-6,-5) for stages 1 through 6 respectively.

6.2 Full tables showing optimized parameters

Here we show additional parameters of the optimized designs for the MISTIE scenario. Tables 6.1, 6.2 and 6.3 show designs from optimizing with no structural restrictions on the alpha allocations. Tables 6.4, 6.5 and 6.6 show results from optimizing over the structured form proposed in the main text.

Stage:	1	2	3	4
H_1	0.000464	0.002682	0.005421	0.003945
H_2	0.001461	0.005204	0.001342	0.000666
H_C	0.001123	0.002040	0.000318	0.000333

Table 6.1: Stage-Specific Alpha Allocations in Unstructured Design Optimized for $\mathscr{H}^{COV}$ (MISTIE)

Stage:	1	2	3	4	5
H_1	0.000682	0.003301	0.002953	0.001482	0.003335
H_2	0.001717	0.002472	0.000540	0.000422	0.001074
H_C	0.001954	0.002341	0.001188	0.000892	0.000646

Table 6.2: Stage-Specific Initial Alpha Allocations in Unstructed Design Optimized for $\mathscr{H}^{MB}$ (MISTIE)

	Reallocation Proportion
g_{12}	0.050
g_{2C}	0.059
g_{C1}	0.261

Table 6.3: Alpha Reallocations in Unstructured Design Optimized for $\mathscr{H}^{MB}$ (MISTIE)

	1	2	3	4	5
H_1	0.000063	0.000687	0.001890	0.003874	0.011057
H_2	0.002328	0.000944	0.000618	0.000515	0.000644
H_C	0.000250	0.000423	0.000441	0.000485	0.000780

Table 6.4: Stage-Specific Alpha Allocations in Structured Design Optimized for $\mathscr{H}^{COV}$ (MISTIE)

	1	2	3	4	5	6
H_1	0.000697	0.002259	0.002174	0.001503	0.004799	0.007198
H_2	0.001097	0.000856	0.000481	0.000263	0.000655	0.000722
H_3	0.001345	0.000356	0.000160	0.000079	0.000180	0.000175

Table 6.5: Stage-Specific Initial Alpha Allocations in Structured Design Optimized for $\mathscr{H}^{MB}$ (MISTIE)

	Reallocation Proportion
g_{12}	0.253
g_{2C}	0.965
g_{C1}	0.555

Table 6.6: Alpha Reallocations in Structured Design Optimized for $\mathscr{H}^{MB}$ (MISTIE)

Bibliography

Ashburner, J. and Friston, K. J. (2000). Voxel-based morphometry—the methods. *Neuroimage*, 11(6):805–821.

Babamoradi, H., van den Berg, F., and Rinnan, Å. (2012). Bootstrap based confidence limits in principal component analysis–a case study. *Chemometrics and Intelligent Laboratory Systems*.

Belia, S., Fidler, F., Williams, J., and Cumming, G. (2005). Researchers misunderstand confidence intervals and standard error bars. *Psychological methods*, 10(4):389.

Bélisle, C. J. (1992). Convergence theorems for a class of simulated annealing algorithms on rd. *Journal of Applied Probability*, pages 885–895.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159.

Beran, R. and Srivastava, M. S. (1985). Bootstrap tests and confidence regions for functions of a covariance matrix. *The Annals of Statistics*, pages 95–115.

Berk, R., Brown, L., and Zhao, L. (2010). Statistical inference after model selection. *Journal of Quantitative Criminology*, 26(2):217–236.

Beyth-Marom, R., Fidler, F., and Cumming, G. (2008). Statistical cognition: Towards evidence-based practice in statistics and statistics education. *Statistics Education Research Journal*, 7(2):20–39.

Bobb, J. F., Schwartz, B. S., Davatzikos, C., and Caffo, B. (2014). Cross-sectional and longitudinal association of body mass index and brain volume. *Human brain mapping*, 35(1):75–88.

Branke, J., Meisel, S., and Schmidt, C. (2008). Simulated annealing in the presence of noise. *Journal of Heuristics*, 14(6):627–654.

Bretz, F., Maurer, W., Brannath, W., and Posch, M. (2009). A graphical approach to sequentially rejective multiple test procedures. *Statistics in Medicine*, 28(4):586.

Calhoun, V., Adali, T., Pearlson, G., and Pekar, J. (2001). A method for making group inferences from functional MRI data using independent component analysis. *Human brain mapping*, 14(3):140–151.

Cassidy, J. (2013). The Reinhart and Rogoff Controversy: A Summing Up. *The New Yorker*. `http://www.newyorker.com/news/john-cassidy/the-reinhart-and-rogoff-controversy-a-summing-up` (accessed August 21, 2014).

Chatterjee, S. (1984). Variance estimation in factor analysis: An application of the bootstrap. *British Journal of Mathematical and Statistical Psychology*, 37(2):252–262.

Cleveland, W., Diaconis, P., and McGill, R. (1982). Variables on scatterplots look more highly correlated when the scales are increased. *Science*, 216.

Cleveland, W. S., McGill, R., et al. (1985). Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833.

Crainiceanu, C. M., Caffo, B. S., Di, C.-Z., and Punjabi, N. M. (2009). Nonparametric signal extraction and measurement error in the analysis of electroencephalographic activity during sleep. *Journal of the American Statistical Association*, 104(486):541–555.

Crainiceanu, C. M., Caffo, B. S., Luo, S., Zipunnikov, V. M., and Punjabi, N. M. (2011). Population value decomposition, a framework for the analysis of image populations. *Journal of the American Statistical Association*, 106(495).

Crainiceanu, C. M., Staicu, A.-M., Ray, S., and Punjabi, N. (2012). Bootstrap-based inference on the difference in the means of two correlated functional processes. *Statistics in Medicine*, 31(26):3223–3240.

Daudin, J., Duby, C., and Trecourt, P. (1988). Stability of principal component analysis studied by the bootstrap method. *Statistics: A Journal of Theoretical and Applied Statistics*, 19(2):241–258.

Davatzikos, C., Genc, A., Xu, D., and Resnick, S. M. (2001). Voxel-based morphometry using the ravens maps: methods and validation using simulated longitudinal atrophy. *NeuroImage*, 14(6):1361–1369.

Di, C.-Z., Crainiceanu, C. M., Caffo, B. S., and Punjabi, N. M. (2009). Multilevel functional principal component analysis. *Annals of Applied Statistics*, 3(1):458–488.

Diaconis, P. and Efron, B. (1983). Computer-intensive methods in statistics. *Scientific American*, 248(5):116–130.

Do, C. B., Chen, Z., Brandman, R., and Koller, D. (2013). Self-Driven Mastery in Massive Open Online Courses. *MOOCs FORUM*, 1:14–16.

Eales, J. D. and Jennison, C. (1992). An improved method for deriving optimal one-sided group sequential tests. *Biometrika*, 79(1):13–24.

Efron, B. (1987). Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185.

Efron, B. and Tibshirani, R. (1993). *An introduction to the bootstrap*, volume 57. CRC press.

Fink, T. M. (1998). *Inverse protein folding, hierarchical optimisation and tie knots*. PhD thesis, University of Cambridge.

Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2012). *Applied longitudinal analysis*, volume 998. John Wiley & Sons.

Gastwirth, J. L. (1988). *Statistical Reasoning in Law and Public Policy: Volume 2: Tort Law, Evidence and Health*, volume 2. Academic Press.

Girshick, M. (1939). On the sampling theory of roots of determinantal equations. *The Annals of Mathematical Statistics*, 10(3):203–224.

Goldsmith, J., Greven, S., and Crainiceanu, C. (2013). Corrected confidence bands for functional data using principal components. *Biometrics*, 69(1):41–51.

Goldszal, A. F., Davatzikos, C., Pham, D. L., Yan, M. X., Bryan, R. N., and Resnick, S. M. (1998). An image-processing system for qualitative and quantitative volumetric analysis of brain images. *Journal of computer assisted tomography*, 22(5):827–837.

Graf, A. C., Posch, M., and Koenig, F. (2015). Adaptive designs for subpopulation analysis optimizing utility functions. *Biometrical Journal*, 57(1):76–89.

Hall, P. and Hosseini-Nasab, M. (2006). On properties of functional principal components analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):109–126.

Hampson, L. V. and Jennison, C. (2013). Group sequential tests for delayed responses (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(1):3–54.

Hampson, L. V. and Jennison, C. (2015). Optimizing the data combination rule for seamless phase ii/iii clinical trials. *Statistics in Medicine*, 34(1):39–58.

Hanley, D. (2012). Mistie phase ii results: Safety, efficacy and surgical performance. International Stroke Conference.

Heer, J. and Bostock, M. (2010). Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 203–212. ACM.

Hong, S., Mitchell, S. K., and Harshman, R. A. (2006). Bootstrap scree tests: A monte carlo simulation and applications to published data. *British Journal of Mathematical and Statistical Psychology*, 59(1):35–57.

Jackson, D. A. (1993). Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. *Ecology*, pages 2204–2214.

Jackson, D. A. (1995). Bootstrapped principal components analysis- reply to mehlman et al. *Ecology*, 76(2):644–645.

Jager, L. R. and Leek, J. T. (2007). Empirical estimates suggest most published medical research is true. *PLoS Med*, 4(4):e168.

Jennison, C. and Turnbull, B. W. (1999). *Group Sequential Methods with Applications to Clinical Trials*. Chapman and Hall/CRC Press.

Johnstone, I. M. (2001). On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pages 295–327.

Jolliffe, I. (2005). *Principal component analysis*. Wiley Online Library.

Jung, S. and Marron, J. (2009). PCA consistency in high dimension, low sample size context. *The Annals of Statistics*, 37(6B):4104–4130.

Koch, I. (2013). *Analysis of Multivariate and High-Dimensional Data*. Cambridge University Press. Cambridge Books Online.

Kollo, T. and Neudecker, H. (1993). Asymptotics of eigenvalues and unit-length eigenvectors of sample variance and correlation matrices. *Journal of Multivariate Analysis*, 47(2):283–300.

Kollo, T. and Neudecker, H. (1997). Asymptotics of pearson-hotelling principal-component vectors of sample variance and correlation matrices. *Behaviormetrika*, 24:51–70.

Krisam, J. and Kieser, M. (2015). Optimal decision rules for biomarker-based subgroup selection for a targeted therapy in oncology. *International Journal of Molecular Sciences*, 16(5):10354.

Kushner, H. (1987). Asymptotic global behavior for stochastic approximation and diffusions with slowly decreasing noise effects: global minimization via monte carlo. *SIAM Journal on Applied Mathematics*, 47(1):169–185.

Lambert, Z. V., Wildt, A. R., and Durand, R. M. (1990). Assessing sampling variation relative to number-of-factors criteria. *Educational and Psychological Measurement*, 50(1):33–48.

Lambert, Z. V., Wildt, A. R., and Durand, R. M. (1991). Approximating confidence intervals for factor loadings. *Multivariate Behavioral Research*, 26(3):421–434.

Ledford, H. (2011). Paper on genetics of longevity retracted. *Nature - News*. `http://www.nature.com/news/2011/110721/full/news.2011.429.html` (accessed August 21, 2014).

Leek, J. T. (2011). Asymptotic conditional singular value decomposition for high-dimensional genomic data. *Biometrics*, 67(2):344–352.

Liu, Q. and Anderson, K. M. (2008). On adaptive extensions of group sequential trials for clinical investigations. *Journal of the American Statistical Association*.

Liyanagunawardena, T. R., Adams, A. A., and Williams, S. A. (2013). MOOCs: A Systematic Study of the Published Literature 2008-2012. *The International Review of Research in Open and Distance Learning*, 14(3):202–227.

Mak, S., Williams, R., and Mackness, J. (2010). Blogs and Forums as Communication and Learning Tools in a MOOC. *University of Lancaster*.

Maryak, J. L. and Chin, D. C. (2001). Global random optimization by simultaneous perturbation stochastic approximation. In *American Control Conference, 2001. Proceedings of the 2001*, volume 2, pages 756–762. IEEE.

Maurer, W. and Bretz, F. (2013). Multiple testing in group sequential trials using graphical approaches. *Statistics in Biopharmaceutical Research*, 5(4):311–320.

Mehlman, D. W., Shepherd, U. L., and Kelt, D. A. (1995). Bootstrapping principal components analysis: a comment. *Ecology*, 76(2):640–643.

Meyer, J. and Shinar, D. (1992). Estimating correlations from scatterplots. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 34(3):335–349.

Milan, L. and Whittaker, J. (1995). Application of the parametric bootstrap to models that incorporate a singular value decomposition. *Applied Statistics*, pages 31–49.

Morgan, T., Zuccarello, M., Narayan, R., Keyl, P., Lane, K., and Hanley, D. (2008). Preliminary findings of the minimally-invasive surgery plus rtpa for intracerebral hemorrhage evacuation (mistie) clinical trial. *Acta Neurochir Suppl.*, 105:147–51.

Nadler, B. (2008). Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, pages 2791–2817.

Nuzzo, R. (2014). Scientific method: Statistical errors. *Nature*, 506(7487):150–152.

O'Brien, P. C. and Fleming, T. R. (1979). A multiple testing procedure for clinical trials. *Biometrics*, pages 549–556.

Ogasawara, H. (2002). Concise formulas for the standard errors of component loading estimates. *Psychometrika*, 67(2):289–297.

Pelley, S. (2012). Deception at Duke: Fraud in Cancer Care? *CBS - 60 Minutes*. `http://www.cbsnews.com/news/deception-at-duke-fraud-in-cancer-care` (accessed August 21, 2014).

Peres-Neto, P. R., Jackson, D. A., and Somers, K. M. (2005). How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis*, 49(4):974–997.

Pocock, S. J. (1977). Group sequential methods in the design and analysis of clinical trials. *Biometrika*, 64:191–199.

Quan, S. F., Howard, B. V., Iber, C., Kiley, J. P., Nieto, F. J., O'Connor, G. T., Rapoport, D. M., Redline, S., Robbins, J., Samet, J., et al. (1997). The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085.

Raykov, T. and Little, T. D. (1999). A note on procrustean rotation in exploratory factor analysis: A computer intensive approach to goodness-of-fit evaluation. *Educational and psychological measurement*, 59(1):47–57.

Rensink, R. A. and Baldridge, G. (2010). The perception of correlation in scatterplots. In *Computer Graphics Forum*, volume 29, pages 1203–1210. Wiley Online Library.

Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects. *Statistical science*, pages 15–32.

Rosenblum, M., Fang, X., and Liu, H. (2014). Optimal, two stage, adaptive enrichment designs for randomized trials using sparse linear programming. *Johns Hopkins University, Dept. of Biostatistics Working Papers. Working Paper 273. http://biostats.bepress.com/jhubiostat/paper273.*

Rosenblum, M., Thompson, R. E., Luber, B. S., and Hanley, D. F. (2015). Adaptive group sequential designs that balance the benefits and risks of expanding inclusion criteria. *Johns Hopkins University, Dept. of Biostatistics Working Papers. Working Paper 250.*

Sadigh-Eteghad, S., Talebi, M., and Farhoudi, M. (2012). Association of apolipoprotein e epsilon 4 allele with sporadic late onset alzheimer's disease. *A meta-analysis. Neurosciences (Riyadh)*, 17(4):321–326.

Salibián-Barrera, M., Van Aelst, S., and Willems, G. (2006). Principal components analysis based on multivariate mm estimators with fast and robust bootstrap. *Journal of the American Statistical Association*, 101(475):1198–1211.

Schwartz, B. S., Caffo, B., Stewart, W. F., Hedlin, H., James, B. D., Yousem, D., and Davatzikos, C. (2010). Evaluation of cumulative lead dose and longitudinal changes in structural mri in former organolead workers. *Journal of occupational and environmental medicine/American College of Occupational and Environmental Medicine*, 52(4):407.

Schwartz, B. S., Chen, S., Caffo, B., Stewart, W. F., Bolla, K. I., Yousem, D., and Davatzikos, C. (2007). Relations of brain volumes with cognitive function in males 45 years and older with past lead exposure. *Neuroimage*, 37(2):633–641.

Schwartz, L. M., Woloshin, S., Black, W. C., and Welch, H. G. (1997). The role of numeracy in understanding the benefit of screening mammography. *Annals of internal medicine*, 127(11):966–972.

Shen, D., Shen, H., and Marron, J. (2012a). A general framework for consistency of principal component analysis. *arXiv preprint arXiv:1211.2671*.

Shen, D., Shen, H., and Marron, J. (2013). Consistency of sparse PCA in high dimension, low sample size contexts. *Journal of Multivariate Analysis*, 115:317–333.

Shen, D., Shen, H., Zhu, H., and Marron, J. (2012b). High dimensional principal component scores and data visualization. *arXiv preprint arXiv:1211.2679*.

Sheridan, S. L., Pignone, M. P., and Lewis, C. L. (2003). A randomized comparison of patients' understanding of number needed to treat and other common risk reduction formats. *Journal of General Internal Medicine*, 18(11):884–892.

Stewart, W., Schwartz, B., Davatzikos, C., Shen, D., Liu, D., Wu, X., Todd, A., Shi, W., Bassett, S., and Youssem, D. (2006). Past adult lead exposure is linked to neurodegeneration measured by brain mri. *Neurology*, 66(10):1476–1484.

Thall, P. F., Simon, R., and Ellenberg, S. S. (1988). Two-stage selection and testing designs for comparative clinical trials. *Biometrika*, 75(2):303–310.

Thompson, B. (1988). Program facstrap: A program that computes bootstrap estimates of factor structure. *Educational and Psychological Measurement*, 48(3):681–686.

Timmerman, M. E., Kiers, H. A., and Smilde, A. K. (2007). Estimating confidence intervals for principal component loadings: A comparison between the bootstrap and asymptotic results. *British Journal of Mathematical and Statistical Psychology*, 60(2):295–314.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.

Tufte, E. R. and Graves-Morris, P. (1983). *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT.

Wang, S. J., Hung, H., and O'Neill, R. T. (2009). Adaptive patient enrichment designs in therapeutic trials. *Biometrical Journal*, 51:358–374.

Wason, J. and Jaki, T. (2012). Optimal design of multi-arm multi-stage trials. *Statistics in Medicine*, 31(30):4269–4279.

Wason, J., Mander, A. P., and Thompson, S. G. (2012). Optimal multistage designs for randomised clinical trials with continuous outcomes. *Statistics in Medicine*, 31(4):301–312.

Zipunnikov, V., Caffo, B., Yousem, D. M., Davatzikos, C., Schwartz, B. S., and Crainiceanu, C. (2011a). Functional principal component model for high-dimensional brain imaging. *NeuroImage*, 58(3):772–784.

Zipunnikov, V., Caffo, B., Yousem, D. M., Davatzikos, C., Schwartz, B. S., and Crainiceanu, C. (2011b). Multilevel functional principal component analysis for high-dimensional data. *Journal of Computational and Graphical Statistics*, 20(4).

Chapter 7

Curriculum Vitae

Education

PhD in Biostatistics (2016), Johns Hopkins Bloomberg School of Public Health
Advisors: Vadim Zipunnikov & Brian Caffo

BA in Economics (2010), University of Rochester

Academic Papers

Peer-Reviewed Publications

M. Rosenblum, T. Qian, Y. Du, H. Qiu, **A. J. Fisher** (2016). Multiple Testing Procedures for Adaptive Enrichment Designs: Combining Group Sequential and Reallocation Approaches. *Biostatistics.* (link).

A. J. Fisher, G. B. Anderson, R. Peng, J. Leek (2014). A randomized trial in a massive online open course shows people don't know what a statistically significant relationship looks like, but they can learn. *PeerJ.* (link; 6,428 unique visitors as of September 1, 2015).

To Appear

Y. Webb-Vargas, S. Chen, **A. J. Fisher,** A. Mejia, Y. Xu, C. Crainiceanu, B. Caffo, M. A. Lindquist (2016). Big Data and Neuroimaging. *Statistics in Biosciences (Invited Submission).*

A. J. Fisher, B. Caffo, B. Schwartz, V. Zipunnikov (accepted in 2015). Fast, Exact Bootstrap Principal Component Analysis for $p > 1$ million. *Journal of the American Statistical Association (TM)*. (link).

Submitted

R. Y. Coley, **A. J. Fisher**, M. Mamawala, H. B. Carter, K. J. Pienta, S. L. Zeger (2015). Bayesian Joint Hierarchical Model for Prediction of Latent Health States with Application to Active Surveillance of Prostate Cancer. (link).

T. Qian, E. Colantuoni, **A. J. Fisher**, M. Rosenblum (2015). Impact of Delayed Outcomes, Accrual Rates, and Prognostic Variables on a Simulated Randomized Trial with Adaptive Enrichment. (link).

A. J. Fisher, H Jaffee, M Rosenblum (2014). interAdapt – An Interactive Tool for Designing and Evaluating Randomized Trials with Adaptive Enrollment Criteria. (link).

Technical Reports

A. J. Fisher, R. Y. Coley, S. L. Zeger (2015). Fast Out-of-Sample Predictions for Bayesian Hierarchical Models of Latent Health States. (link).

Awards and Scholarships

Margaret Merrell Award (2016, co-winner with Amanda Mejia): Departmental award recognizing outstanding research by a Biostatistics doctoral student

June B. Culley Award (2014): Honors outstanding achievement by a Biostatistics student on his or her school-wide oral examination paper

Doctoral Training Grant in Environmental Biostatistics (2012-2015): Provides funding for at least three years

Undergraduate Awards (2006-2010): Phi Beta Kappa; John Dows Mairs Prize (University of Rochester Economics Dept); Omicron Delta Epsilon International Honor Society for Economics; Theta Chi Long, Walter, Ott Award; Theta Chi Valentine H. Zahn Fund

Software

bootSVD: An R package for implementing fast, exact bootstrap principal component analysis and singular value decompositions for high dimensional data (i.e. $>$ 1 million covariates). Matrices too large for memory can be entered as class `ff` objects, with contents stored on disk. (CRAN link; GitHub link)

ggBrain: An R package for beautiful brain image figures with ggplot. This packages allows color to be mapped to both (1) tissue intensities of the template image, and (2) values of a voxel-wise test statistic. (GitHub link)

interAdapt: An interactive tool for designing and evaluating randomized trials with adaptive enrollment criteria (Shiny App link; CRAN link; Github link)

Computer Skills

Advanced Skills: R

Basic Skills: git, Python, MATLAB, D3.js, stata, C, shell scripting, LaTeX

Reviewer

Journal of the American Statistical Association (2015)

Risk Analysis (2014)

Teaching

Co-Instructor

Statistical Reasoning I and II (2015): My role included teaching independently for 13 hours of lectures *(MPH Level Course, JHSPH Summer Institute of Epidemiology and Biostatistics)*

Guest Lecturer

Essentials of Probability and Statistical Inference I-II (2013) *(Biostatistics ScM Level, JHSPH)*

Lab Lecturer with Content Design

Essentials of Probability and Statistical Inference I-IV (2012-2014): Designed and administered a weekly 1-hour lab lecture. In the second year of this course, we reduced this lab to a 1-hour session every two weeks. *(Biostatistics ScM Level, JHSPH)*

Lab Lecturer without Content Design

Statistical Methods in Public Health II (2014-2015): Administered approximately 16 hours of lab lecture in each year of the course. *(MPH Level, JHSPH)*

Educational Presentations

JHU Biostatistics Journal Club (2013-2015): I have given talks on high dimensional asymptotics, adaptive clinical trials, and the Bayesian Bootstrap

JHU Biostatistics Computing Club (2013-2015): I have given talks on environments in R, and on LaTeX

General TA Roles

Statistical Methods in Public Health I and IV (2014-2015) *(MPH Level, JHSPH)*

Statistical Reasoning I and II (2012), *(MPH Level, JHSPH Summer Institute of Epidemiology and Biostatistics)*

Conference Presentations

"A Randomized Trial in a Massive Online Open Course Shows People Don't Know What a Statistically Significant Relationship Looks Like, but They Can Learn." JSM 2015, Seattle WA. *Contributed Speed Session & Poster.*

"Fast Exact Bootstrap Principal Component Analysis for $p > 1$ million." ENAR 2015, Miami, FL. *Contributed Talk.*

"Fast, Exact Bootstrap Principal Component Analysis for $p > 1$ Million." 4th Annual Hopkins Imaging Conference 2014 (link). Baltimore, MD, *Invited Short Talk & Poster.*

"Fast Exact Bootstrap Principal Component Analysis for $p > 1$ million: Leveraging Low-Dimensional Structure Across High-Dimensional Bootstrap Samples." JSM 2014, Boston, MA. *Contributed Speed Session & Poster.*

"People Can't See Statistical Significance: A Massive Randomized Trial on the Visual Perception of Relationships." ENAR 2014, Baltimore, MD, *Contributed Talk.*

Other Leadership & Service Roles

Volunteer with Thread (2015)

Facilitator at JHU Data Science Hackathon (2015): Assisted a team through the process of scraping web data and building a shiny app (3-day event)

Co-organizer of JHU Biostatistics Computing Club (2012-2013), with Prasad Patil (speaker schedule link)