LARGE-SCALE STATISTICAL LEARNING METHODS AND ALGORITHMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Junyang Qian

May 2020

Preface

The past two decades have witnessed rapid growth in the amount of data available to us. Many areas such as genomics, neuroscience, economics and Internet services are producing big datasets that have high dimension, large sample size, or both. This provides unprecedented opportunities for us to retrieve and infer valuable information from the data. Meanwhile, it also poses new challenges for statistical methodologies and computational algorithms.

Over the years, there has been prosperous development on both sides. In high-dimensional regression problems, it is often believed that among all the predictors, only a subset of them are relevant to the prediction, also known as the "bet on sparsity" principle. It is both of scientific interests and for practical performance to identify such a subset from the data and establish a sparse model. The lasso is a widely used and effective method of simultaneous estimation and variable selection for linear models and has been studied to have nice theoretical properties under certain conditions. Random forest, boosting and multivariate adaptive regression splines are representative nonparametric methods for nonlinear modeling. In the meantime, a variety of efficient algorithms, implementations and computational tools have been developed to accommodate the need for fast computation. For practical applications, on the one hand, we want to formulate a reasonable model to capture the desired structures and improve the quality of statistical estimation and inference. On the other hand, in the face of increasingly large datasets, computation can be a big hurdle for one to arrive at meaningful conclusions. This thesis stands at the intersection of the two topics, proposing statistical methods to capture desired structures in the data, and seeking scalable approaches to optimizing the computation for very large datasets.

Among the others, genomic data often present the nature of ultrahigh dimensionality. Researchers used to deal with wide data in such studies, where the number of variables was large but

the sample size was fairly limited. One can still conduct somewhat sophisticated statistical analyses in memory and within a reasonable amount of time. However, recent studies have collected genetic and disease information from very large cohorts. For example, the UK Biobank genotypes and phenotypes dataset contains about 500,000 individuals and more than 800,000 genotyped SNP measurements per person, the size of which may well exceed the physical memory we have. The computational challenges for very large datasets are thus two fold. For implementation, we need to deal with the fact that we may not hold the entire data in memory, which is often an assumption made by most statistical packages, including the highly efficient lasso package **glmnet**. For the design of algorithms, we need to take into consideration not only the complexity of basic operations but also the cost associated with disk I/O – data transfer between the memory and the disk — a very expensive operation that is several magnitudes slower than in-memory operations. To this end, in Chapter 1, we design a scalable iterative algorithm called BASIL that solves the lasso and elastic-net efficiently on large-scale and ultrahigh-dimensional data. The implementation for SNP data in PLINK2 format is packaged in an R package **snpnet**. We demonstrate the method on the UK Biobank and see strong performance gain over the other methods.

In some applications, we are interested in predicting more than one response with the same set of predictors and the responses are correlated in some way, and we want to leverage such structure to further improve the statistical efficiency and predictive performance. This falls into the topic of multivariate statistics, and also known as multitask learning. In particular, one typical correlation structure we want to model is low-rank structure for the linear model. We assume that the true responses are not only linear in the predictors but also the linear coefficient matrix is low-rank. In other words, the dependency of the responses on the predictors is through a layer of shared components constructed linearly from the predictors. For high dimensional problems, it is also important to assume sparse effect since again, it is unlikely that all the predictors are relevant to the prediction. One regression method with these two joint structures is known as Sparse Reduced Rank Regression. In Chapter 2, we investigated into this regression method, extending it to dealing with missing values, confounding covariates, and design a scalable algorithm based on a screening mechanism. This method is implemented in an R package **multiSnpnet**. We demonstrate the method on the UK Biobank and see improved performance over other methods including the lasso.

In Chapter 3, we switch gears and study a different class of problems — estimating the heterogeneous treatment effect. This type of problems often arises when we are faced with several options

and would like to determine the optimal one. For example, a doctor may have two treatments and want to decide on one depending on the patient's characteristics and symptoms. An e-commerce company may have several ads to choose from for a placeholder and want to determine one that will bring it the most profit based on the user group. This falls into the topic of causal inference, and in particular the estimation of heterogeneous treatment effect. The main challenge of such problems is that in the historical data, we never observe the other side of the coin, so we have no access to the ground truth of the true difference between these options at all. There can also be redundant predictors that we don't know beforehand. We exploit two classic nonparametric methods gradient boosting and multivariate adaptive regression splines, and adapt them to the context of causal inference to estimate the treatment effect based on the predictors available. The two methods show fairly competitive performance compared with other methods. The implementation is packaged in an R package **causalLearning**.

Acknowledgments

First and foremost, I would like to thank my advisor Trevor Hastie. Working with Trevor has shown me his wisdom and insight in both statistics and computation. His passion for practical methodologies that have impact in the broader scientific communities and constant pursuit of understanding have shaped a great deal of my academic altitude. Trevor is always supportive, encouraging and considerate, and is always willing to devote his time and effort to the growth of his students — to be both a good scholar and a good person. I could not have asked for a better mentor.

Next I would like to thank Robert Tibshirani, for his countless guidance and advice on the research throughout the entire thesis, and Manuel Rivas, for embarking on the statistical genetics research and his patience, guidance and continued encouragement during this process. These projects would not be possible without their support.

I am very grateful to Jonathan Taylor and Stefan Wager for agreeing to devote their time and effort to be on my committee, to Bradley Efron, Balasubramanian Narasimhan and Julia Palacios for their advice on these research projects, and to Benjamin Van Roy for introducing me to the field of reinforcement learning and getting me exposure to interesting projects. I am extremely fortunate to be in the Statistical Learning group where I have learned a great deal from Trevor, Rob, Jonathan and all the fellow students over the years. I am also very grateful to the Rivas Lab, Yosuke Tanigawa, Ruilin Li, Matthew Aguirre and Chris Chang for fruitful discussion and patiently sharing their expertise on quantitative genetics and computational techniques. I would like to thank the Two Sigma Graduate Fellowship for supporting my graduate studies.

I am very thankful to all the friends who made my journey at Stanford so enjoyable. I would like to thank Rakesh Achanta, Joey Arthur, Nan Bi, Zhou Fan, Jessica Hwang, Jiyao Kou, Snigdha

Panigrahi, Kris Sankaran, Subhabrata Sen for spending together a wonderful first year and being friends always good to chat with. Special thanks to Chen Liang, Michael Zhu, Ciyang Qing, Chen Chen and Minda Deng for a lot of food tasting and adventures, to Chaojun Wang, Charles Zheng for random fun chat and games at Sequoia, and to Yan Min, Jason Lee, Chi Jin, Junzi Zhang and Song Mei for a lot of helpful advice and discussions. I am also very grateful to have Weijie Su, Camile Pataki, Scott Powers, Wenfei Du and Linxi Liu for great collaboration and friendship.

Finally, I want to thank my parents, for their unconditional support, and for their belief in whatever I choose to do.

Contents

Preface **iv**

Acknowledgments **vii**

1 Fast Sparse Regression for Large-scale and Ultrahigh-dimensional Problems **1**

1.1 Introduction 1

1.2 Results 4

1.3 Discussion 22

1.4 Materials and Methods 24

1.5 Appendix: Manhattan Plots 34

2 Fast Reduced Rank Regression for Large-scale and Ultrahigh-dimensional Problems **37**

2.1 Introduction 37

2.1.1 Reduced-Rank Regression for Multiple Responses 40

2.1.2 Sparse Models in High-Dimensional Problems 41

2.2 Sparse Reduced-Rank Regression 42

2.3 Fast Algorithms for Large-Scale and Ultrahigh-Dimensional Problems 43

2.3.1 Alternating Minimization 43

2.3.2 Variable Screening for Ultrahigh-Dimensional Problems 45

2.3.3 Computational Considerations 49

2.3.4 Extensions 50

2.3.5 Full Algorithm 55

2.4 Convergence Analysis . . . 57
2.5 Simulation Studies . . . 58
2.6 Real Data Application: UK Biobank . . . 60
2.6.1 Asthma and 7 Blood Biomarkers . . . 61
2.6.2 35 Biomarkers . . . 63
2.7 Related Work . . . 66
2.8 Summary and Discussion . . . 67
2.9 Appendix . . . 69
2.9.1 Additional Proofs . . . 69
2.9.2 Connection with CCA . . . 71
2.9.3 Additional Experiments . . . 73
2.9.4 Additional Information on the Methods . . . 79

3 Some Methods for Heterogeneous Treatment Effect Estimation in High Dimensions 82

3.1 Introduction . . . 82
3.2 Related work . . . 84
3.2.1 Propensity score methods . . . 85
3.3 Transformed outcome regression and conditional mean regression . . . 87
3.3.1 Shared-basis conditional mean regression . . . 90
3.4 Pollinated transformed outcome (PTO) forests . . . 90
3.5 Causal boosting . . . 92
3.5.1 Cross-validation for causal boosting . . . 94
3.5.2 Within-leaf propensity adjustment . . . 95
3.6 Causal MARS . . . 97
3.6.1 Confidence intervals . . . 99
3.7 Simulation study . . . 101
3.8 Application . . . 107
3.9 Discussion . . . 112
3.10 Appendix . . . 112

Bibliography 114

List of Tables

1.1 R^2 values for height. For sequential LR, lasso and relaxed lasso, the chosen model is based on maximum R^2 on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column. The validation results for PRS-CS and SBayesR are not available because we used a combined training and validation set for training. 14

1.2 Comparison of prediction results on height with the model trained following the same procedure as ours except for an additional prescreening step as done in Lello et al. [2018]. In addition to R^2, proportion of residual variance explained (denoted by h^2_{test}) and correlation between the fitted values and actual values are computed. We also compute an adjusted correlation between the residual after regressing age and sex out from the prediction and the residual after regressing age and sex out from the true response, both on the test set. 16

1.3 R^2 values for BMI. For lasso and relaxed lasso, the chosen model is based on maximum R^2 on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column. 18

1.4 AUC values for asthma. For lasso and relaxed lasso, the chosen model is based on maximum AUC on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column. 19

1.5 AUC values for high cholesterol. For lasso and relaxed lasso, the chosen model is based on maximum AUC on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column. 21

1.6 Timing performance (milliseconds) on multiplication of SNP matrix and residual matrix. The methods are all implemented in C++ and run on a Macbook with 2.9 GHz Intel Core i7 and 8 GB 1600 MHz DDR3. 29

1.7 Timing comparison on a synthetic dataset of size $n = 50,000$ and $p = 100,000$. The time for **bigstatsr** and **biglasso** has two components: one for the conversion to the desired data type and the other for the actual computation. The experiments are all run with 16 cores and 64 GB memory. 33

3.1 Specifications for the 16 simulation scenarios. The four rows of the table correspond, respectively, to the sample size, dimensionality, mean effect function, treatment effect function and noise level. Simulations 1 through 8 use randomized treatment assignment, meaning $\pi(x) = 1/2$. Simulations 9 through 16 have a bias in treatment assignment, specified by (3.4). 104

List of Figures

1.1 The lasso coefficient profile that shows the progression of the BASIL algorithm. The previously finished part of the path is colored grey, the newly completed and verified is in green, and the part that is newly computed but failed the verification is colored red. 5

1.2 Comparison of different methods on the test set. R^2 are evaluated for continuous phenotypes height and BMI, and AUC evaluated for binary phenotypes asthma and high cholesterol. 13

1.3 R^2 plot for height. The top axis shows the number of active variables in the model. . 15

1.4 Left: actual height versus predicted height on 5000 random samples from the test set. The correlation between actual height and predicted height is 0.9416. Right: histogram of the lasso residuals for height. Standard deviation of the residual is 5.05 (cm). 15

1.5 Comparison of the lasso coefficients and univariate p-values for height. The index on the horizontal axis represents the SNPs sorted by their univariate p-values. The red curve associated with the left vertical axis shows the $-\log_{10}$ of the univariate p-values. The blue bars associated with the right vertical axis show the corresponding lasso coefficients for each (sorted) SNP. The horizontal dotted lines in gray identifies lasso coefficients of ± 0.05. The vertical one represents the 100K cutoff used in Lello et al. [2018]. 17

1.6 R^2 plot for BMI. The top axis shows the number of active variables in the model. . . 18

1.7 Left: actual BMI versus predicted BMI on 5000 random samples from the test set. The correlation between actual BMI and predicted BMI is 0.3256. Right: residuals of lasso prediction for BMI. Standard deviation of the residual is 4.51 kg/m^2. 19
1.8 AUC plot for asthma. The top axis shows the number of active variables in the model. 20
1.9 Results for asthma based on the best lasso model. Left: box plot of the percentile of the linear prediction score among cases versus controls. Right: the stratified prevalence across different percentile bins based on the predicted scores by the optimal lasso. 20
1.10 AUC plot for high cholesterol. The top axis shows the number of active variables in the model. 21
1.11 Results for high cholesterol based on the best lasso model. Left: box plot of the percentile of the linear prediction score among cases versus controls. Right: the stratified prevalence across different percentile bins based on the predicted scores by the optimal lasso. 22
1.12 ROC curves. Left: asthma. Right: high cholesterol. 22
1.13 Histogram of the standard deviations of the SNPs. They are computed *after* mean imputation of the missing values because they would be the exact standardization factors to be used if the lasso were applied with variable standardization on the mean-imputed SNP matrix. 27
1.14 Manhattan plots of the univariate p-values and lasso coefficients for height (a, c) and BMI (b, d). The vertical axis of the p-value plots shows $-\log_{10}(p)$ for each SNP, while the vertical axis of the coefficient plots shows the magnitude of the coefficients from **snpnet**. The SNPs with relatively large lasso coefficients are highlighted in green. The red horizontal line on the p-value plot represents a reference level of $p = 5 \times 10^{-8}$. 35
1.15 Manhattan plots of the univariate p-values and lasso coefficients for asthma (a, c) and high cholesterol (b, d). The vertical axis of the p-value plots shows $-\log_{10}(p)$ for each SNP, while the vertical axis of the coefficient plots shows the magnitude of the coefficients from **snpnet**. The SNPs with relatively large lasso coefficients are highlighted in green. The red horizontal line on the p-value plot represents a reference level of $p = 5 \times 10^{-8}$. 36

2.1 Diagram of the reduced rank regression. The nodes in grey are latent variables. The arrows represent the dependency structure. 41
2.2 Size of screened set under different strategies. Left: independent design. Right: equi-correlated design with $\rho = 0.5$. Signal-to-noise ratio (SNR) = 1, and we use the true rank = 3. 48
2.3 R^2 each run is evaluated on a test set of size 5000. "oracle" is the result where we know the true active variables and solve on this subset of variables. "glmnet" fits the responses separately. "SRRR-r" indicates the SRRR results with assumed rank r. . . 59
2.4 Asthma and Basophil count prediction performance plots. Different colors correspond to lower rank predictive performance across (x-axis) training data set and (y-axis) validation data set for (left) asthma and (right) basophil count. 62
2.5 Change in prediction accuracy for multiresponse model compared to single response model. (top) (y-axis 1 bar) R^2 relative change (%) for each phenotype (x-axis) and R^2 absolute change (y-axis 2). 63
2.6 Alanine aminotransferase and albumin prediction performance plots. Different colors correspond to lower rank predictive performance across (x-axis) training data set and (y-axis) validation data set for (left) alanine aminotransferase and (right) albumin. For lower rank representation we applied lazy rank evaluation. 64
2.7 Change in prediction accuracy for multiresponse model compared to single response model. (top) (y-axis 1 bar) R^2 relative change (%) for each biomarker (x-axis) across different biomarker category (color) and R^2 absolute change (y-axis 2). (bottom) Change in predictive accuracy for multiresponse model compared to single response model for urinary biomarkers. 65

2.8 The latent structures of the the top five key SRRR components for AST to ALT ratio. Using trait squared cosine score described in Tanigawa et al. [2019], the top five key SRRR components for AST to ALT ratio (components 9, 18, 20, 8, and 3) are identified from a full-rank SVD of coefficient matrix C from SRRR ($C = UDV^T$) and shown as a series of biplots. In each panel, principal components of genetic variants (rows of UD) are shown in blue as scatter plot using the main axis and singular vectors of traits (rows of V) are shown in red dots with lines using the secondary axis, for the identified key components. The five traits and variants with the largest distance from the center of origin are annotated with their name. 66
2.9 Scenario 1-9. R^2 each run is evaluated on a test set of size 5000. 75
2.10 Scenario 10-18. R^2 each run is evaluated on a test set of size 5000. The oracle here does not take into account the noise in true coefficient matrix, and do reduced rank regression on the true support and the true rank. 76
2.11 Scenario 19-27. R^2 each run is evaluated on a test set of size 5000. 77
2.12 Scenario 28-36. R^2 each run is evaluated on a test set of size 5000. 78
2.13 The identification of unrelated White British individuals in UK Biobank. The first two genotype principal components (PCs) are shown on the x- and y-axis and the identified unrelated White British individuals (Methods) are shown in red. 81

3.1 The variance of two ATE estimators for $n = 10$, 30, 100 and 300, as the ratio of the absolute main effect $|\mu_1 + \mu_0|/2$ to the noise level σ increases from 0 to 0.5. 89
3.2 A comparison of raw and pollinated transformed outcome forests. Each method is applied to a randomized simulation and a non-randomized simulation, and we visually compare the estimated treatment effect with the true treatment effect We see that in each case, the pollination improves the estimates. 92

3.3 Confidence intervals for causal forest in Scenario 8 from Section 3.7. On the left in blue we plot the true treatment effect for each patient against the index of the patient, sorted by treatment effect. The thin gray lines show the average upper and lower confidence interval bounds for each patient, and the dotted black line smooths over these averages. On the left the thin lines give the miscoverage rate for each patient, and the thick lines smooth over these thin lines. These results reflect 100 simulations using 50 bagged causal trees. 100

3.4 Confidence intervals for causal MARS in Scenario 8 from Section 3.7. On the left in blue we plot the true treatment effect for each patient against the index of the patient, sorted by treatment effect. The thin gray lines show the average upper and lower confidence interval bounds for each patient across 100 simulations, and the dotted black line smooths over these averages. On the left the thin lines give the miscoverage rate for each patient, and the thick lines smooth over these thin lines. These results reflect 100 simulations using 50 bagged causal MARS models. 101

3.5 Bias-corrected confidence intervals for causal MARS in Scenario 8 from Section 3.7. On the left in blue we plot the true treatment effect for each patient against the index of the patient, sorted by treatment effect. The thin gray lines show the average upper and lower confidence interval bounds for each patient across 100 simulations, and the dotted black line smooths over these averages. On the left the thin lines give the miscoverage rate for each patient, and the thick lines smooth over these thin lines. These results reflect 100 simulations using 50 bagged causal MARS models. 102

3.6 Results across eight simulated randomized experiments. For details of the generating distributions, see Table 3.1. The seven estimators being evaluated are: NULL = the null prediction, GF = gradient forest, PTO0 = pollinated transformed outcome forest (using propensity = 1/2), CB0 = causal boosting, CM0 = causal MARS. The vertical blue bar shows the standard deviation of the response, for assessing the practical significance of the difference between the methods' performances. 105

3.7 Results across eight simulated observational studies, in which treatment is more likely to be assigned to those with a greater mean effect. The seven estimators being evaluated are: NULL = the null prediction, GF = gradient forest, PTO = pollinated transformed outcome forest, CB1 = causal boosting (propensity adjusted), CB0 = causal boosting, CM1 = causal MARS (propensity adjusted), CM0 = causal MARS. CB0 and CM0 are in gray because they would not be used in this setting. They are provided for reference to assess the effect of the propensity adjustment. The vertical blue bar shows the standard deviation of the response, for assessing the practical significance of the difference between the methods' performances. 106

3.8 Personalized treatment effect estimates from causal boosting and causal MARS. Each circle represents a patient, who gets a personalized estimate from each method. The dashed line represents the diagonal, along which the two estimates are the same. . . 108

3.9 Decision trees summarizing with broad strokes the inferences of causal boosting and causal MARS. The variables are: `trr` triglcerides (mg/dL) from blood draw; `age` (years) age at beginning of trial; `glur` glucose (mg/dL) from blood draw; `screat` creatinine (mg/dL) from blood draw; `umalcr` albumin/creatinine ratio from urine sample; `dbp` diastolic blood pressure (mm Hg); `egfr` estimated glomerular filtration rate (mL/min/1.73m^2). If the inequality at a split is true for a patient, then that patient belongs to the left daughter node. 109

3.10 Training set personalized treatment effects, estimated via causal boosting, versus estimated glomerular filtration rate (eGFR). Patients are stratified according to eGFR on the x-axis, and each point gives the average personalized treatment effect among patients in that stratum. Error bars correspond to one standard error for the mean personalized treatment effect. The vertical dashed line represents a medical cutoff, below which patients are considered to suffer from chronic kidney disease. 110

3.11 Validation set personalized treatment effects, estimated via causal boosting, versus estimated glomerular filtration rate (eGFR). Patients are stratified according to eGFR on the x-axis, and each point gives the average personalized treatment effect among patients in that stratum. Error bars correspond to one standard error for the mean personalized treatment effect. The vertical dashed line represents a medical cutoff, below which patients are considered to suffer from chronic kidney disease. . . 111

Chapter 1

Fast Sparse Regression for Large-scale and Ultrahigh-dimensional Problems

1.1 Introduction

The past two decades have witnessed rapid growth in the amount of data available to us. Many areas such as genomics, neuroscience, economics and Internet services are producing big datasets that have high dimension, large sample size, or both. A variety of statistical methods and computing tools have been developed to accommodate this change. See, for example, Friedman et al. [2009], Efron and Hastie [2016], Dean and Ghemawat [2008], Zaharia et al. [2010], Abadi et al. [2016] and the references therein for more details.

In high-dimensional regression problems, we have a large number of predictors, and it is likely that only a subset of them have a relationship with the response and will be useful for prediction. Identifying such a subset is desirable for both scientific interests and the ability to predict outcomes in the future. The lasso [Tibshirani, 1996] is a widely used and effective method for simultaneous estimation and variable selection. Given a continuous response $y \in \mathbb{R}^n$ and a model matrix $X \in$

$\mathbb{R}^{n\times p}$, it solves the following regularized regression problem.

$$\hat{\beta}(\lambda) = \text{argmin}_{\beta\in\mathbb{R}^p} \ \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1, \tag{1.1}$$

where $\|x\|_q = (\sum_{i=1}^n |x_i|^q)^{1/q}$ is the vector ℓ_q norm of $x \in \mathbb{R}^n$ and $\lambda \geq 0$ is the tuning parameter. The ℓ_1 penalty on β allows for selection as well as estimation. Normally there is an unpenalized intercept in the model, but for ease of presentation we leave it out, or we may assume that both X and y have been centered with mean 0. One typically solves the entire lasso solution path over a grid of λ values $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_L$ and chooses the best λ by cross-validation or by predictive performance on an independent validation set. In R [R Core Team, 2017], several packages, such as **glmnet** [Friedman et al., 2010b] and **ncvreg** [Breheny and Huang, 2011], provide efficient procedures to obtain the solution path for the Gaussian model (1.1), and for other generalized linear models with the residual sum of squared replaced by the negative log-likelihood of the corresponding model. Among them, **glmnet**, equipped with highly optimized Fortran subroutines, is widely considered the fastest off-the-shelf lasso solver. It can, for example, fit a sequence of 100 logistic regression models on a sparse dataset with 54 million samples and 7 million predictors within only 2 hours [Hastie, 2015].

However, as the data become increasingly large, many existing methods and tools may not be able to serve the need, especially if the size exceeds the memory size. Most packages, including the ones mentioned above, assume that the data or at least its sparse representation can be fully loaded in memory and that the remaining memory is sufficient to hold other intermediate results. This becomes a real bottleneck for big datasets. For example, in our motivating application, the UK Biobank genotypes and phenotypes dataset [Bycroft et al., 2018] contains about 500,000 individuals and more than 800,000 genotyped single nucleotide polymorphisms (SNPs) measurements per person. This provides unprecedented opportunities to explore more comprehensive genotypic relationships with phenotypes of interest. For polygenic traits such as height and body mass index (BMI), specific variants discovered by genome-wide association studies (GWAS) used to explain only a small proportion of the estimated heritability [Visscher et al., 2017], an upper bound of the proportion of phenotypic variance explained by the genetic components. While GWAS with larger sample size on the UK Biobank can be used to detect more SNPs and rare variants, their prediction performance is fairly limited by univariate models. It is very interesting to see if full-scale multiple

regression methods such as the lasso or elastic-net can improve the prediction performance and simultaneously select relevant variants for the phenotypes. That being said, the computational challenges are two fold. First is the memory bound. Even though each bi-allelic SNP value can be represented by only two bits and the **PLINK** library [Chang et al., 2015] stores such SNP datasets in a binary compressed format, statistical packages such as **glmnet** and **ncvreg** require that the data be loaded in memory in a normal double-precision format. Given its sample size and dimension, the genotype matrix itself will take up around one terabyte of space, which may well exceed the size of the memory available and is infeasible for the packages. Second is the efficiency bound. For a larger-than-RAM dataset, it has to sit on the disk and we may only read part of it into the memory. In such scenario, the overall efficiency of the algorithm is not only determined by the number of basic arithmetic operations but also the disk I/O — data transfer between the memory and the disk — an operation several magnitudes slower than in-memory operations.

In this paper, we propose an efficient and scalable meta algorithm for the lasso called Batch Screening Iterative Lasso (BASIL) that is applicable to larger-than-RAM datasets and designed to tackle the memory and efficiency bound. It computes the entire lasso path and can easily build on any existing package to make it a scalable solution. As the name suggests, it is done in an iterative fashion on an adaptively screened subset of variables. At each iteration, we exploit an efficient, parallelizable screening operation to significantly reduce the problem to one of manageable size, solve the resulting smaller lasso problem, and then reconstruct and validate a full solution through another efficient, parallelizable step. In other words, the iterations have a screen-solve-check substructure. That being said, it is the goal and also the guarantee of the BASIL algorithm that the final solution exactly solves the full lasso problem (1.1) rather than any approximation, even if the intermediate steps work repeatedly on subsets of variables.

The screen-solve-check substructure is inspired by Tibshirani et al. [2012] and especially the proposed strong rules. The strong rules state: assume $\hat{\beta}(\lambda_{k-1})$ is the lasso solution in (1.1) at λ_{k-1}, then the jth predictor is discarded at λ_k if

$$|x_j^\top (y - X\hat{\beta}(\lambda_{k-1}))| < \lambda_k - (\lambda_{k-1} - \lambda_k). \tag{1.2}$$

The key idea is that the inner product above is almost "non-expansive" in λ and that the lasso solution is characterized equivalently by the Karush-Kuhn-Tucker (KKT) condition [Boyd and

Vandenberghe, 2004]. For the lasso, the KKT condition states that $\hat{\beta} \in \mathbb{R}^p$ is a solution to (1.1) if for all $1 \leq j \leq p$,

$$\frac{1}{n} \cdot x_j^\top (y - X\hat{\beta}) \begin{cases} = \lambda \cdot \text{sign}(\hat{\beta}_j), \text{ if } \hat{\beta}_j \neq 0, \\ \leq \lambda, \text{ if } \hat{\beta}_j = 0. \end{cases} \tag{1.3}$$

The KKT condition suggests that the variables discarded based on the strong rules would have coefficient 0 at the next λ_k. The checking step comes into play because this is not a guarantee. The strong rules can fail, though failures occur rarely when $p > n$. In any case, the KKT condition will be checked to see if the coefficients of the left-out variables are indeed 0 at λ_k. If the check fails, we add in the violated variables and repeat the process. Otherwise, we successfully reconstruct a full solution and move to the next λ. This is the iterative algorithm proposed by these authors and has been implemented efficienly into the **glmnet** package.

The BASIL algorithm proceeds in a similar way but is designed to optimize for datasets that are too big to fit into the memory. Considering the fact that screening and KKT check need to scan through the entire data and are thus costly in the disk Input/Output (I/O) operations, we attempt to do batch screening and solve *a series of* models (at different λ values) in each iteration, where a single sweep over the full data would suffice. Followed by a checking step, we can obtain the lasso solution for multiple λ's in one iteration. This can effectively reduce the total number of iterations needed to compute the full solution path and thus reduce the expensive disk read operations that often cause significant delay in the computation. The process is illustrated in Figure 1.1 and will be detailed in the next section.

1.2 Results

Overview of the BASIL algorithm For convenience, we first introduce some notation. Let $\Omega = \{1, 2, \ldots, p\}$ be the universe of variable indices. For $1 \leq \ell \leq L$, let $\hat{\beta}(\lambda_\ell)$ be the lasso solution at $\lambda = \lambda_\ell$, and $\mathcal{A}(\lambda_\ell) = \{1 \leq j \leq p : \hat{\beta}_j(\lambda_\ell) \neq 0\}$ be the active set. When X is a matrix, we use $X_\mathcal{S}$ to represent the submatrix including only columns indexed by $\mathcal{S}$. Similarly when β is a vector, $\beta_\mathcal{S}$ represents the subvector including only elements indexed by $\mathcal{S}$. Given any two vectors $a, b \in \mathbb{R}^n$, the dot product or inner product can be written as $a^\top b = \langle a, b \rangle = \sum_{i=1}^n a_i b_i$. Throughout the paper, we use predictors, features, variables and variants interchangeably. We use the strong set to refer to the screened subset of variables on which the lasso fit is computed at each iteration, and

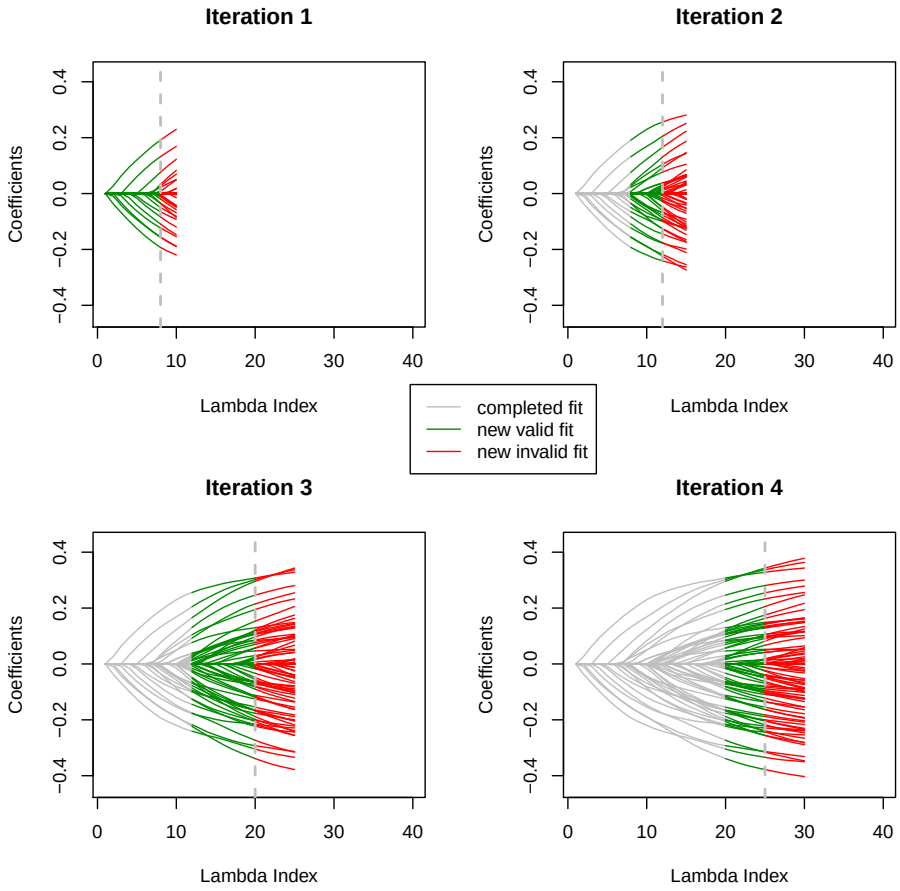


Figure 1.1: The lasso coefficient profile that shows the progression of the BASIL algorithm. The previously finished part of the path is colored grey, the newly completed and verified is in green, and the part that is newly computed but failed the verification is colored red.

the active set to refer to the subset of variables with nonzero lasso coefficients.

Remember that our goal is to compute the exact lasso solution (1.1) for larger-than-RAM datasets over a grid of regularization parameters $\lambda_1 > \lambda_2 > \cdots > \lambda_L \geq 0$. We describe the procedure for the Gaussian family in this section and discuss extension to general problems in the next. A common choice is $L = 100$ and $\lambda_1 = \max_{1 \leq j \leq p} |x_j^\top r^{(0)}|/n$, the largest λ at which the estimated coefficients start to deviate from zero. Here $r^{(0)} = y$ if we do not include an intercept term and $r^{(0)} = y - \bar{y}$ if we do. In general, $r^{(0)}$ is the residual of regressing y on the unpenalized variables, if any. The other λ's can be determined, for example, by an equally spaced array on

the log scale. The solution path is found iteratively with a screening-solving-checking substructure similar to the one proposed in Tibshirani et al. [2012]. Designed for large-scale and ultrahigh-dimensional data, the BASIL algorithm can be viewed as a batch version of the strong rules. At each iteration we attempt to find valid lasso solution for *multiple* λ values on the path and thus reduce the burden of disk reads of the big dataset. Specifically, as summarized in Algorithm 1, we start with an empty strong set $\mathcal{S}^{(0)} = \emptyset$ and active set $\mathcal{A}^{(0)} = \emptyset$. Each of the following iterations consists of three steps: screening, fitting and checking.

Algorithm 1 BASIL for the Gaussian Model

1: **Initialization**: active set $\mathcal{A}^{(0)} = \emptyset$, initial residual $r^{(0)}$ (with respect to the intercept or other unpenalized variables) at $\lambda_1 = \lambda_{\max}$, a short list of initial parameters $\Lambda^{(0)} = \{\lambda_1, \ldots, \lambda_{L^{(0)}}\}$.

2: **for** $k = 0$ **to** K **do**

3: **Screening**: for each $1 \le j \le p$, compute inner product with current residual $c_j^{(k)} = \langle x_j, r^{(k)} \rangle$. Construct the strong set

$$\mathcal{S}^{(k)} = \mathcal{A}^{(k)} \cup \mathcal{E}_M^{(k)},$$

where $\mathcal{E}_M^{(k)}$ is the set of M variables in $\Omega \setminus \mathcal{A}^{(k)}$ with largest $|c^{(k)}|$.

4: **Fitting**: for all $\lambda \in \Lambda^{(k)}$, solve the lasso only on the strong set $\mathcal{S}^{(k)}$, and find the coefficients $\hat{\beta}^{(k)}(\lambda)$ and the residuals $r^{(k)}(\lambda)$.

5: **Checking**: search for the smallest λ such that the KKT conditions are satisfied, i.e.,

$$\bar{\lambda}^{(k)} = \min \left\{ \lambda \in \Lambda^{(k)} : \max_{j \in \Omega \setminus \mathcal{S}^{(k)}} (1/n) |x_j^\top r^{(k)}(\lambda)| < \lambda \right\}.$$

For empty set, we define $\bar{\lambda}^{(k)}$ to be the immediate previous λ to $\Lambda^{(k)}$ but increment M by ΔM. Let the current active set $\mathcal{A}^{(k+1)}$ and residuals $r^{(k+1)}$ defined by the solution at $\bar{\lambda}^{(k)}$. Define the next parameter list $\Lambda^{(k+1)} = \{\lambda \in \Lambda^{(k)} : \lambda < \bar{\lambda}^{(k)}\}$. Extend this list if it consists of too few elements. For $\lambda \in \Lambda^{(k)} \setminus \Lambda^{(k+1)}$, we obtain exact lasso solutions for the full problem:

$$\hat{\beta}_{\mathcal{S}^{(k)}}(\lambda) = \hat{\beta}^{(k)}(\lambda), \quad \hat{\beta}_{\Omega \setminus \mathcal{S}^{(k)}}(\lambda) = 0.$$

6: **end for**

In the screening step, an updated strong set is found as the candidate for the subsequent fitting. Suppose that so far (valid) lasso solutions have been found for $\lambda_1, \ldots, \lambda_\ell$ but not for $\lambda_{\ell+1}$. The new set will be based on the lasso solution at λ_ℓ. In particular, we will select the top M variables with largest absolute inner products $|\langle x_j, y - X\hat{\beta}(\lambda_\ell)|$. They are the variables that are most likely to be active in the lasso model for the next λ values. In addition, we include the ever-active variables at $\lambda_1, \ldots, \lambda_\ell$ because they have been "important" variables and might continue to be important at a later stage.

In the fitting step, the lasso is fit on the updated strong set for the next λ values $\lambda_{\ell+1}, \ldots, \lambda_{\ell'}$. Here ℓ' is often smaller than L because we do not have to solve for all of the remaining λ values on this strong set. The full lasso solutions at much smaller λ's are very likely to have active variables outside of the current strong set. In other words even if we were to compute solutions for those very small λ values on the current strong set, they would probably fail the KKT test. These λ's are left to later iterations when the strong set is expanded.

In the checking step, we check if the newly obtained solutions on the strong set can be valid part of the full solutions by evaluating the KKT condition. Given a solution $\hat{\beta}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ to the sub-problem at λ, if we can verify for every left-out variable j that $(1/n)|\langle x_j, y - X_{\mathcal{S}}\hat{\beta}_{\mathcal{S}}\rangle| < \lambda$, we can then safely set their coefficients to 0. The full lasso solution $\hat{\beta}(\lambda) \in \mathbb{R}^p$ is then assembled by letting $\hat{\beta}_{\mathcal{S}}(\lambda) = \hat{\beta}_{\mathcal{S}}$ and $\hat{\beta}_{\Omega\setminus\mathcal{S}}(\lambda) = 0$. We look for the λ value prior to the one that causes the first failure down the λ sequence and use its residual as the basis for the next screening. Nevertheless, there is still chance that none of the solutions on the current strong set passes the KKT check for the λ subsequence considered in this iterations. That suggests the number of previously added variables in the current iteration was not sufficient. In this case, we are unable to move forward along the λ sequence, but will fall back to the λ value where the strong set was last updated and include ΔM more variables based on the sorted absolute inner product.

The three steps above can be applied repeatedly to roll out the complete lasso solution path for the original problem. However, if our goal is choosing the best model along the path, we can stop fitting once an optimal model is found evidenced by the performance on a validation set. At a high level, we run the iterative procedure on the training data, monitor the error on the validation set, and stop when the model starts to overfit, or in other words, when the validation error shows a clear upward trend.

Extension to general problems It is straightforward to extend the algorithm from the Gaussian case to more general problems. In fact, the only changes we need to make are the screening step and the strong set update step. Wherever the strong rules can be applied, we have a corresponding version of the iterative algorithm. In Tibshirani et al. [2012], the general problem is

$$\hat{\beta}(\lambda) = \text{argmin}_{\beta\in\mathbb{R}^p} \;\; f(\beta) + \lambda \sum_{j=1}^{r} c_j \|\beta_j\|_{p_j}, \tag{1.4}$$

where f is a convex differentiable function, and for all $1 \leq j \leq r$, $c_j \geq 0, p_j \geq 1$, and β_j can be a scalar or vector whose ℓ_{p_j}-norm is represented by $\|\beta_j\|_{p_j}$. The general strong rule discards predictor j if

$$\|\nabla_j f(\hat{\beta}(\lambda_{k-1}))\|_{q_j} < c_j(2\lambda_k - \lambda_{k-1}), \tag{1.5}$$

where $1/p_j + 1/q_j = 1$. Hence, our algorithm can adapt and screen by choosing variables with large values of $\|\nabla_j f(\hat{\beta}(\lambda_{k-1}))\|_{q_j}$ that are not in the current active set. We expand in more detail two important applications of the general rule: logistic regression and Cox's proportional hazards model in survival analysis.

Logistic regression In the lasso penalized logistic regression [Friedman et al., 2010a] where the observed outcome $y \in \{0,1\}^n$, the convex differential function in (1.4) is

$$f(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log p_i + (1-y_i)\log(1-p_i)\right).$$

where $p_i = 1/(1+\exp(-x_i^\top \beta))$ for all $1 \leq i \leq n$. The rule in (1.5) is reduced to

$$|x_j^\top (y - \hat{p}(\lambda_{k-1}))| < \lambda_k - (\lambda_{k-1} - \lambda_k),$$

where $\hat{p}(\lambda_{k-1})$ is the predicted probabilities at $\lambda = \lambda_{k-1}$. Similar to the Gaussian case, we can still fit relaxed lasso and allow adjustment covariates in the model to adjust for confounding effect.

Cox's proportional hazards model In the usual survival analysis framework, for each sample, in addition to the predictors $x_i \in \mathbb{R}^p$ and the observed time y_i, there is an associated right-censoring indicator $\delta_i \in \{0,1\}$ such that $\delta_i = 0$ if failure and $\delta_i = 1$ if right-censored. Let $t_1 < t_2 < ... < t_m$ be the increasing list of unique failure times, and $j(i)$ denote the index of the observation failing at time t_i. The Cox's proportional hazards model [Cox, 1972] assumes the hazard for the ith individual as $h_i(t) = h_0(t)\exp(x_i^\top \beta)$ where $h_0(t)$ is a shared baseline hazard at time t. We can let $f(\beta)$ be the negative log partial likelihood in (1.4) and screen based on its gradient at the most recent lasso

solution as suggested in (1.5). In particular,

$$f(\beta) = -\frac{1}{m}\sum_{i=1}^{m}\left(x_{j(i)}^\top\beta - \log\left(\sum_{j\in R_i}\exp(x_j^\top\beta)\right)\right),$$

where R_i is the set of indices j with $y_j \geq t_i$ (those at risk at time t_i). We can derive the associated rule based on (1.5) and thus the survival BASIL algorithm. Further discussion and comprehensive experiments are included in a follow-up paper [Li et al., 2020].

Extension to the elastic net Our discussion so far focuses solely on the lasso penalty, which aims to achieve a rather sparse set of linear coefficients. In spite of good performance in many high-dimensional settings, it has limitations. For example, when there is a group of highly correlated variables, the lasso will often pick out one of them and ignore the others. This poses some hardness in interpretation. Also, under high-correlation structure like that, it has been empirically observed that when the predictors are highly correlated, the ridge can often outperform the lasso [Tibshirani, 1996].

The elastic net, proposed in Zou and Hastie [2005], extends the lasso and tries to find a sweet spot between the lasso and the ridge penalty. It can capture the grouping effect of highly correlated variables and sometimes perform better than both methods especially when the number of variables is much larger than the number of samples. In particular, instead of imposing the ℓ_1 penalty, the elastic net solves the following regularized regression problem.

$$\hat{\beta}(\lambda) = \text{argmin}_{\beta\in\mathbb{R}^p}\ \ f(\beta) + \lambda(\alpha\|\beta\|_1 + (1-\alpha)\|\beta\|_2^2/2), \tag{1.6}$$

where the mixing parameter $\alpha \in [0,1]$ determines the proportion of lasso and ridge in the penalty term.

It is straightforward to adapt the BASIL procedure to the elastic net. It follows from the gradient motivation of the strong rules and KKT condition of convex optimization. We take the Gaussian family as an example. The others are similar. In the screening step, it is easy to derive that we can still rank *among the currently inactive variables* on their absolute inner product with the residual $|x_j^\top(y - X\hat{\beta}(\lambda_{k-1}))|$ to determine the next candidate set. In the checking step, to verify that all the left-out variables indeed have zero coefficients, we need to make sure that $(1/n)|x_j^\top(y - X\hat{\beta}(\lambda_{k-1}))| \leq$

$\lambda\alpha$ holds for all such variables. It turns out that in our UK Biobank applications, the elastic-net results (after selection of α and λ on the validation set) do not differ significantly from the lasso results, which will be immediately seen in the next section.

UK Biobank analysis We describe a real-data application on the UK Biobank that in fact motivates our development of the BASIL algorithm.

The UK Biobank [Bycroft et al., 2018] is a very large, prospective population-based cohort study with individuals collected from multiple sites across the United Kingdom. It contains extensive genotypic and phenotypic detail such as genomewide genotyping, questionnaires and physical measures for a wide range of health-related outcomes for over 500,000 participants, who were aged 40-69 years when recruited in 2006-2010. In this study, we are interested in the relationship between an individual's genotype and his/her phenotypic outcome. While GWAS focus on identifying SNPs that may be marginally associated with the outcome using univariate tests, we would like to find relevant SNPs in a multivariate prediction model using the lasso. A recent study [Lello et al., 2018] fits the lasso on a subset of the variables after one-shot univariate p-value screening and suggests improvement in explaining the variation in the phenotypes. However, the left-out variants with relatively weak marginal association may still provide additional predictive power in a multiple regression environment. The BASIL algorithm enables us to fit the lasso model at full scale and gives further improvement in the explained variance over the alternative models considered.

We focused on 337,199 White British unrelated individuals out of the full set of over 500,000 from the UK Biobank dataset [Bycroft et al., 2018] that satisfy the same set of population stratification criteria as in DeBoever et al. [2018]. The dataset is partitioned randomly into training, validation and test subsets. Each individual has up to 805,426 measured variants, and each variant is encoded by one of the four levels where 0 corresponds to homozygous major alleles, 1 to heterozygous alleles, 2 to homozygous minor alleles and NA to a missing genotype. In addition, we have available covariates such as age, sex, and forty pre-computed principal components of the SNP matrix.

To evaluate the predictive performance for quantitative response, we use a common measure R-squared (R^2). Given a linear estimator $\hat{\beta}$ and data (y, X), it is defined as

$$R^2 = 1 - \frac{\|y - X\hat{\beta}\|_2^2}{\|y - \bar{y}\|_2^2}.$$

We evaluate this criteria for all the training, validation and test sets. For a dichotomous response, misclassification error could be used but it would depend on the calibration. Instead the receiver operating characteristic (ROC) curve provides more information and illustrates the tradeoff between true positive and false positive rates under different thresholds. The AUC computes the area under the ROC curve — a larger value indicates a generally better classifier. Therefore, we will evaluate AUCs on the training, validation and test sets for dichotomous responses.

We compare the performance of the lasso with related methods to have a sense of the contribution of different components. Starting from the baseline, we fit a linear model that includes only age and sex (Model 1 in the tables below), and then one that includes additionally the top 10 principal components (Model 2). These are the adjustment covariates used in our main lasso fitting and we use these two models to highlight the contribution of the SNP information over and above that of age, sex and the top 10 PCs. In addition, the strongest univariate model is also evaluated (Model 3). This includes the 12 adjustment covariates together with the single SNP that is most correlated with the outcome after adjustment.

Toward multivariate models, we first compare with a univariate method that has some multivariate flavor (Models 4 and 5). We select a subset of the K most marginally significant variants (after adjusting for the covariates), and construct a new variable by linearly combining these variants using their univariate coefficients. An OLS is then fit on the new variable together with the adjustment variables. It is similar to a one-step partial least squares [Wold, 1975] with p-value based truncation. We take $K = 10,000$ and $100,000$ in the experiments. We further compare with a hierarchical sequence of multivariate models where each is fit on a subset of the most significant SNPs. In particular, the ℓ-th model selects $\ell \times 1000$ SNPs with the smallest univariate p-values, and a multivariate linear or logistic regression is fit on those variants jointly. The sequence of models are evaluated on the validation set, and the one with the smallest validation error is chosen. We call this method Sequential LR or SeqLR (Model 6) for convenience in the rest of the paper. As a byproduct of the lasso, the relaxed lasso [Meinshausen, 2007] fits a debiased model by refitting an OLS on the variables selected by the lasso. This can potentially recover some of the bias introduced by lasso shrinkage. For the elastic-net, we fit separate solution paths with varying λ's at $\alpha = 0.1, 0.5, 0.9$, and evaluate their performance (R^2 or AUC) on the validation set. The best pair of hyperparameters (α, λ) is selected and the corresponding test performance is reported.

In addition, we make comparison with two other Bayesian methods PRS-CS [Ge et al., 2019]

and SBayesR [Lloyd-Jones et al., 2019]. For PRS-CS, we first characterized the GWAS summary statistics using the combined set of training and validation set ($n = 269,927$) with age, sex, and the top 10 PCs as covariates using PLINK v2.00a3LM (9 Apr 2020) [Chang et al., 2015]. Using the LD reference dataset precomputed for the European Ancestry using the 1000 genome samples (`https://github.com/getian107/PRScs`), we applied PRS-CS with the default option. We took the posterior effect size estimates and computed the polygenic risk scores using PLINK2's `--score` subcommand [Chang et al., 2015]. For SBayesR, we computed the sparse LD matrix using the combined set of training and validation set individuals ($n = 269,927$) using the `-- make-sparse-ldm` subcommand implemented in GCTB version 2.0.1 [Zeng et al., 2018]. Using the GWAS summary statistics computed on the set of individuals and following the GCTB's recommendations, we applied SBayesR with the following options: `gctb --sbayes R--ldm [the LD matrix] --pi 0.95,0.02,0.02,0.01 --gamma 0.0,0.01,0.1,1 --chain-length 10000 --burn-in 2000 --exclude-mhc --gwas-summary [the GWAS summary statistics]`. We report the model performance on the test set.

There are thousands of measured phenotypes in the dataset. For demonstration purpose, we analyze four phenotypes that are known to be highly or moderately heritable and polygenic. For these complex traits, univariate studies may not find SNPs with smaller effects, but the lasso model may include them and predict the phenotype better. We look at two quantitative traits: standing height and body mass index (BMI) [Tanigawa et al., 2019], and two qualitative traits: asthma and high cholesterol (HC) [DeBoever et al., 2018].

We first summarize the test performance of different methods on the four phenotypes in Figure 1.2. The lasso and elastic net show significant improvement in test R^2 and AUC over the other competing methods. Details for the four phenotypes are given in the next section. A comparison of the univariate p-values and the lasso coefficients for all these traits is shown in the form of Manhattan plots in the Appendix 1.5 (Supplementary Figure 1.14, 1.15).

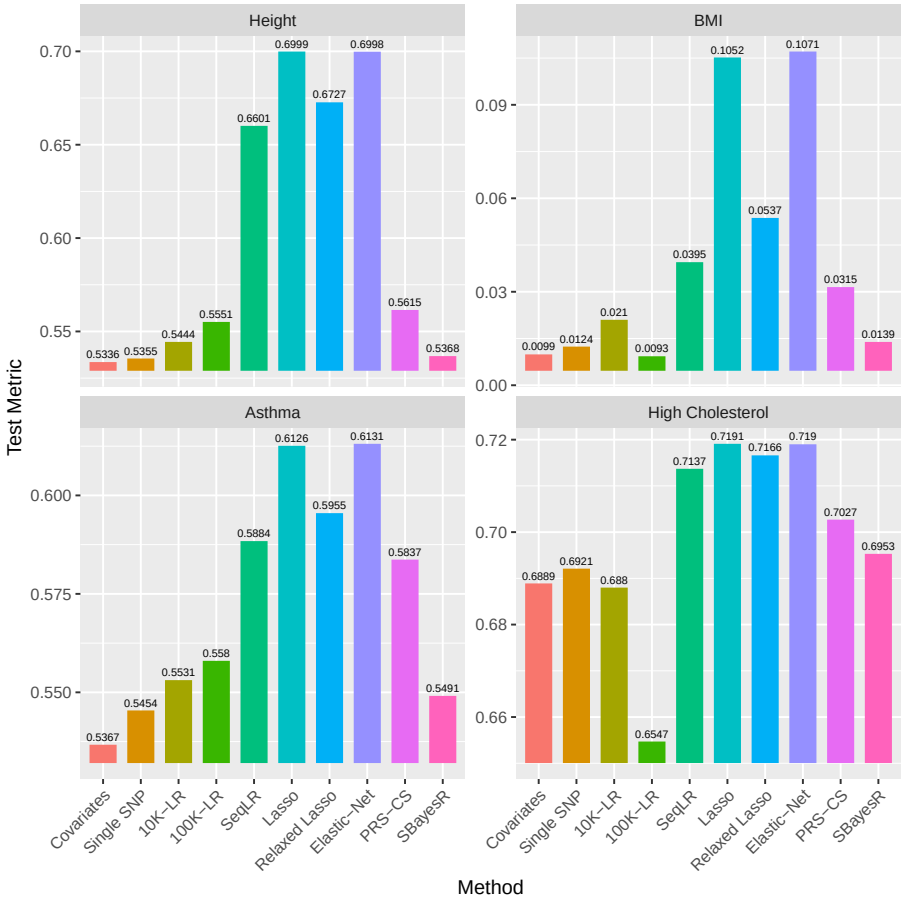


Figure 1.2: Comparison of different methods on the test set. R^2 are evaluated for continuous phenotypes height and BMI, and AUC evaluated for binary phenotypes asthma and high cholesterol.

Standing Height Height is a polygenic and heritable trait that has been studied for a long time. It has been used as a model for other quantitative traits, since it is easy to measure reliably. From twin and sibling studies, the narrow sense heritability is estimated to be 70-80% [Silventoinen et al., 2003, Visscher et al., 2006, 2010]. Recent estimates controlling for shared environmental factors present in twin studies calculate heritability at 0.69 [Zaitlen et al., 2013, Hemani et al., 2013]. A linear based model with common SNPs explains 45% of the variance [Yang et al., 2010] and a model including imputed variants explains 56% of the variance, almost matching the estimated heritability [Yang et al., 2015]. So far, GWAS studies have discovered 697 associated variants that explain one

fifth of the heritability [Lango Allen et al., 2010, Wood et al., 2014]. Recently, a large sample study was able to identify more variants with low frequencies that are associated with height [Marouli et al., 2017]. Using lasso with the larger UK Biobank dataset allows both a better estimate of the proportion of variance that can be explained by genomic predictors and simultaneous selection of SNPs that may be associated. The results are summarized in Table 1.1. The associated R^2 curves for the lasso and the relaxed lasso are shown in Figure 1.3. The residuals of the optimal lasso prediction are plotted in Figure 1.4.

Model	Form	R^2_{train}	R^2_{val}	R^2_{test}	Size
(1)	Age + Sex	0.5300	0.5260	0.5288	2
(2)	Age + Sex + 10 PCs	0.5344	0.5304	0.5336	12
(3)	Strong Single SNP	0.5364	0.5323	0.5355	13
(4)	10K Combined	0.5482	0.5408	0.5444	10,012
(5)	100K Combined	0.5833	0.5515	0.5551	100,012
(6)	Sequential LR	0.7416	0.6596	0.6601	17,012
(7)	Lasso	0.8304	0.6992	**0.6999**	47,673
(8)	Relaxed Lasso	0.7789	0.6718	0.6727	13,395
(9)	Elastic Net	0.8282	0.6991	0.6998	48,256
(10)	PRS-CS	0.5692	–	0.5615	148,052
(11)	SBayesR	0.5397	–	0.5368	667,045

Table 1.1: R^2 values for height. For sequential LR, lasso and relaxed lasso, the chosen model is based on maximum R^2 on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column. The validation results for PRS-CS and SBayesR are not available because we used a combined training and validation set for training.

A large number (47,673) of SNPs need to be selected in order to achieve the optimal $R^2_{\text{test}} = 0.6999$ for the lasso and similarly for the elastic-net. Comparatively, the relaxed lasso sacrifices some predictive performance by including a much smaller subset of variables (13,395). Past the optimal point, the additional variance introduced by refitting such large models may be larger than the reduction in bias. The large models confirm the extreme polygenicity of standing height.

In comparison to the other models, the lasso performs significantly better in terms of R^2_{test} than all univariate methods, and outperforms multivariate methods based on univariate p-value ordering. That demonstrates the value of simultaneous variable selection and estimation from a multivariate perspective, and enables us to predict height to within 10 cm about 95% of the time based only on SNP information (together with age and sex). We also notice that the sequential linear regression approach does a good job, whose performance gets close to that of the relaxed lasso. It is straightforward and easy to implement using existing softwares such as **PLINK** [Chang

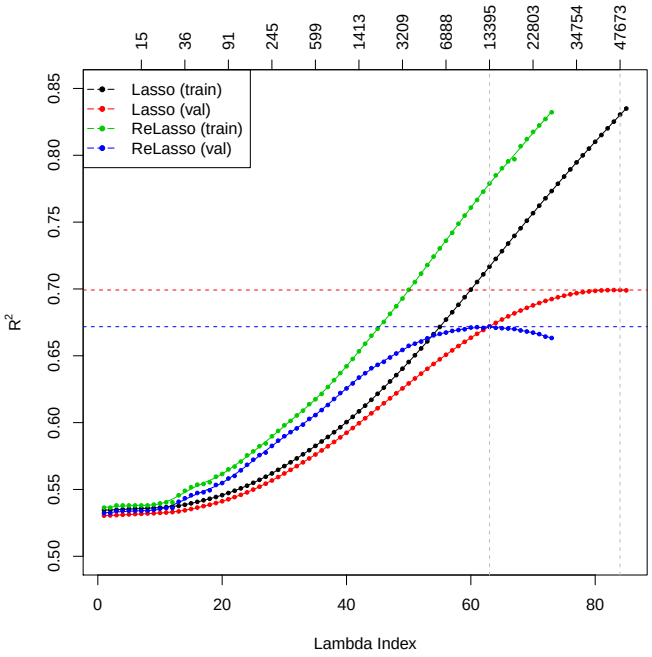


Figure 1.3: R^2 plot for height. The top axis shows the number of active variables in the model.

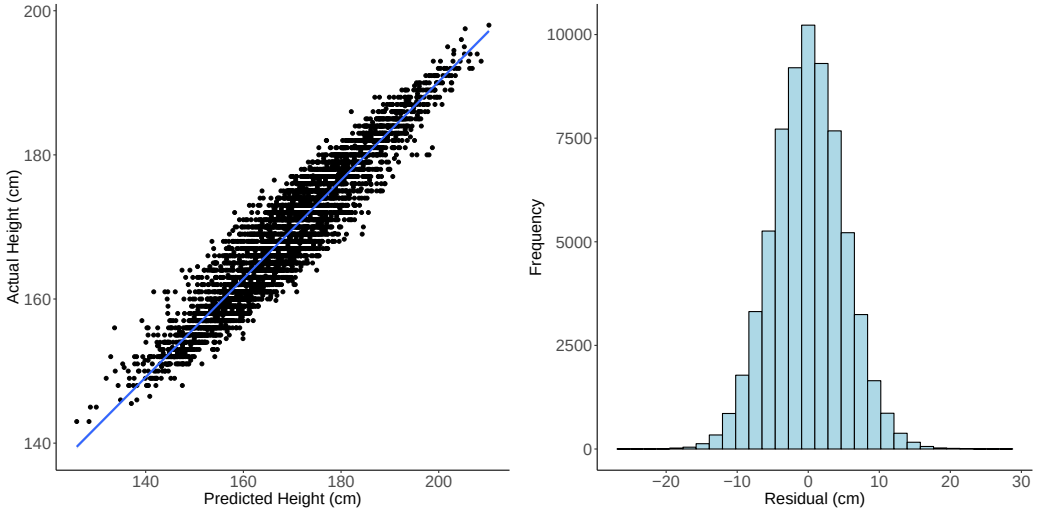


Figure 1.4: Left: actual height versus predicted height on 5000 random samples from the test set. The correlation between actual height and predicted height is 0.9416. Right: histogram of the lasso residuals for height. Standard deviation of the residual is 5.05 (cm).

Method	R^2_{val}	R^2_{test}	h^2_{test}	Cor_{test}	$\text{Cor}_{\text{test}}-\{\text{age, sex}\}$
Lasso	69.92%	69.99%	35.66%	0.8366	0.4079
Prescreened lasso	69.40%	69.56%	34.73%	0.8340	0.4025

Table 1.2: Comparison of prediction results on height with the model trained following the same procedure as ours except for an additional prescreening step as done in Lello et al. [2018]. In addition to R^2, proportion of residual variance explained (denoted by h^2_{test}) and correlation between the fitted values and actual values are computed. We also compute an adjusted correlation between the residual after regressing age and sex out from the prediction and the residual after regressing age and sex out from the true response, both on the test set.

et al., 2015].

Recently Lello et al. [2018] apply a lasso based method to predict height and other phenotypes on the UK Biobank. Instead of fitting on all QC-satisfied SNPs (as stated in Section 1.4), they pre-screen 50K or 100K most significant SNPs in terms of p-value and apply lasso on that set only. In addition, although both datasets come from the same UK Biobank, the subset of individuals they used is larger than ours. While we restrict the analysis to the unrelated individuals who have self-reported white British ancestry, they look at Europeans including British, Irish and Any Other White. For a fair comparison, we follow their procedure (pre-screening 100K SNPs) but run on our subset of the dataset. The results are shown in Table 1.2. We see that the improvement of the full lasso over the prescreened lasso is almost 0.5% in test R^2, and 1% relative to the proportion of residual variance explained after covariate adjustment.

Further, we compare the full lasso coefficients and the univariate p-values from GWAS in Figure 1.5. The vertical grey dotted line indicates the top 100K cutoff in terms of p-value. We see although a general decreasing trend appears in the magnitude of the lasso coefficients with respect to increasing p-values (decreasing $-\log_{10}(p)$), there are a number of spikes even in the large p-value region which is considered marginally insignificant. This shows that variants beyond the strongest univariate ones contribute to prediction.

Body Mass Index (BMI) BMI is another polygenic trait that is widely studied. Like height, it is heritable and easily measured. It is also a trait of interest, since obesity is a risk factor for diseases such as type 2 diabetes and cardiovasclar disease. Recent studies estimate heritability at 0.42 [Zaitlen et al., 2013, Hemani et al., 2013] and 27% of the variance can be explained using a genomic model [Yang et al., 2015]. We expect the heritability to be lower than that for height, since intuitively speaking, one component of the body mass, weight, should heavily depend on

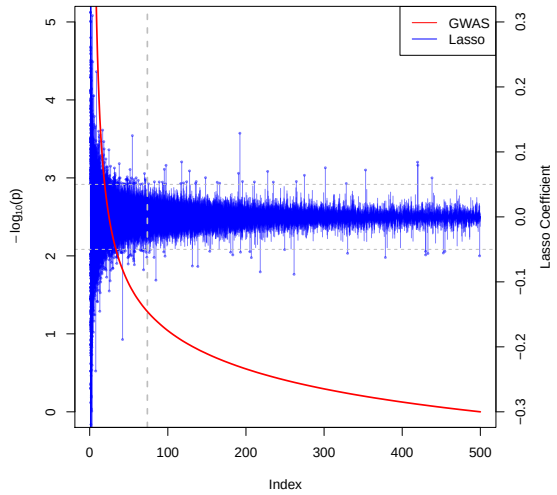


Figure 1.5: Comparison of the lasso coefficients and univariate p-values for height. The index on the horizontal axis represents the SNPs sorted by their univariate p-values. The red curve associated with the left vertical axis shows the $-\log_{10}$ of the univariate p-values. The blue bars associated with the right vertical axis show the corresponding lasso coefficients for each (sorted) SNP. The horizontal dotted lines in gray identifies lasso coefficients of ± 0.05. The vertical one represents the 100K cutoff used in Lello et al. [2018].

environmental factors, for example, individual's lifestyle. From GWAS studies, 97 associated loci have been identified, but they only account for 2.7% of the variance [Speliotes et al., 2010, Locke et al., 2015]. Although the estimates of heritability are not precise, there may be more missing heritability for BMI than for height. We also find lower R^2 values using the lasso. The results are summarized in Table 1.3. The R^2 curves for the lasso and the relaxed lasso are shown in Figure 1.6. From the table, we see that more than 26,000 variants are selected by the lasso to attain an R^2 greater than 10%. In constrast, the relaxed lasso and the sequential linear regression use around one-tenths of the variables, and end up with degraded predictive performance both at around 5%. From Figure 1.7, we see further evidence that the actual BMI is of high variability and hard to predict with the lasso model — the correlation between the predicted value and the actual value is 0.3256. From the residual histogram on the right, we also see the distribution is skewed to the right, suggesting a number of exceedingly high observed values than the ones predicted by the model. Nevertheless, we are able to predict BMI within 9 kg/m^2 about 95% of the time.

Model	Form	R^2_{train}	R^2_{val}	R^2_{test}	Size
(1)	Age + Sex	0.0092	0.0089	0.0083	2
(2)	Age + Sex + 10 PCs	0.0104	0.0103	0.0099	12
(3)	(2) + Single SNP	0.0134	0.0128	0.0124	13
(4)	(2) + 10K Combined	0.0384	0.0195	0.0210	10,012
(5)	(2) + 100K Combined	0.1307	0.0064	0.0093	100,012
(6)	Sequential LR	0.0865	0.0385	0.0395	2,012
(7)	Lasso	0.3196	0.1017	**0.1052**	26,060
(8)	Relaxed Lasso	0.1609	0.0504	0.0537	2,585

Table 1.3: R^2 values for BMI. For lasso and relaxed lasso, the chosen model is based on maximum R^2 on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column.

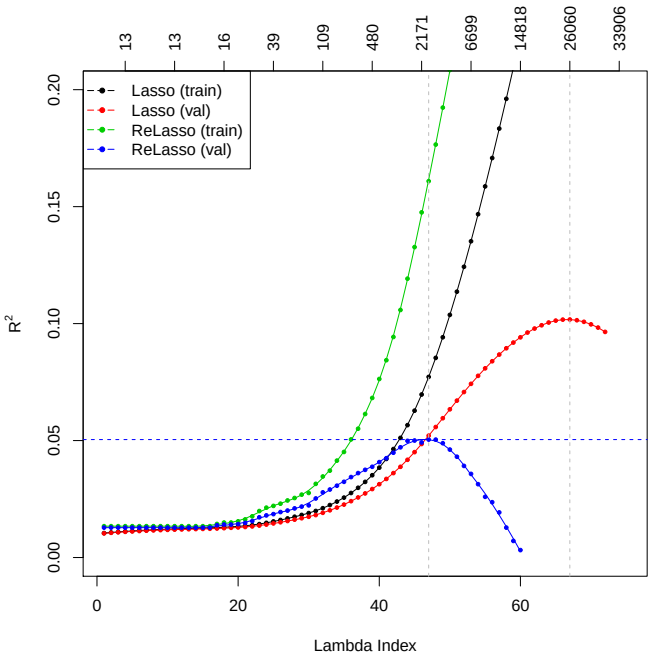


Figure 1.6: R^2 plot for BMI. The top axis shows the number of active variables in the model.

Asthma Asthma is a common respiratory disease characterized by inflammation of airways in the lungs and difficulty breathing. It is another complex, polygenic trait that is associated with both genetic and environmental factors. Our results are summarized in Table 1.4. The AUC curves for the lasso and the relaxed lasso are shown in Figure 1.8. In addition, for each test sample, we compute the percentile of its predicted score/probability among the entire test cohort, and create box plots of such percentiles separately for the control group and the case group. We see on the left

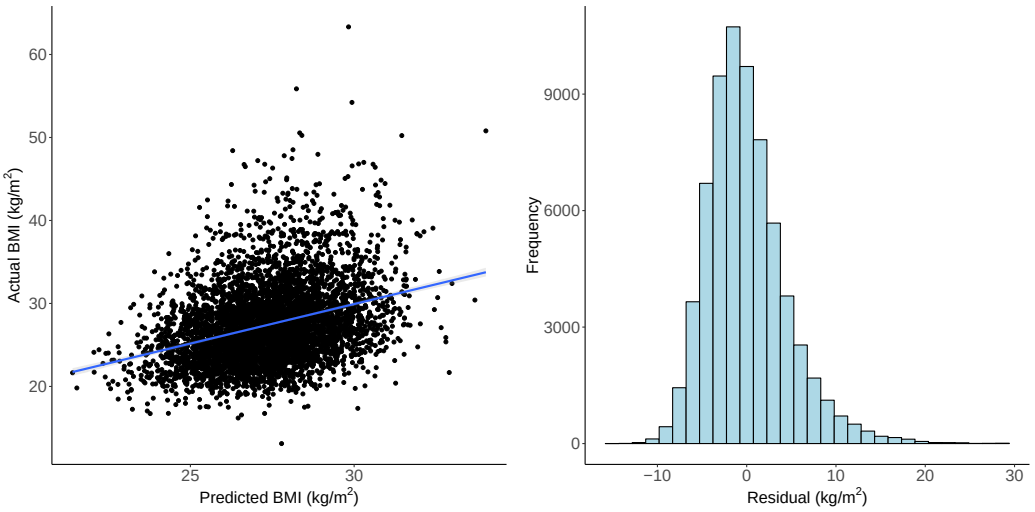


Figure 1.7: Left: actual BMI versus predicted BMI on 5000 random samples from the test set. The correlation between actual BMI and predicted BMI is 0.3256. Right: residuals of lasso prediction for BMI. Standard deviation of the residual is 4.51 kg/m^2.

Model	Form	$\text{AUC}_{\text{train}}$	AUC_{val}	AUC_{test}	Size
(1)	Age + Sex	0.5293	0.5297	0.5320	2
(2)	Age + Sex + 10 PCs	0.5342	0.5344	0.5367	12
(3)	(2) + Single SNP	0.5463	0.5476	0.5454	13
(4)	(2) + 10K Combined	0.5783	0.5580	0.5531	10,012
(5)	(2) + 100K Combined	0.6884	0.5644	0.5580	100,012
(6)	Sequential LR	0.6601	0.5883	0.5884	2,012
(7)	Lasso	0.7692	0.6159	**0.6126**	5,936
(8)	Relaxed Lasso	0.6747	0.5988	0.5955	621

Table 1.4: AUC values for asthma. For lasso and relaxed lasso, the chosen model is based on maximum AUC on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column.

of Figure 1.9 that there is a significant overlap between the box plots of the two groups, suggesting that asthma is difficult to predict. This can also be seen from the AUC value and the ROC curve in Figure 1.12. That being said, the multivariate lasso still does much better than the baseline model and the strongest univariate model. On the right of Figure 1.9, we stratify the prediction percentile into 10 bins, and compute the overall prevalence within each bin. We observe a clear upward trend that provides further evidence that we manage to capture some genetic signal there.

High Cholesterol High cholesterol is characterized by high amounts of cholesterol present in the

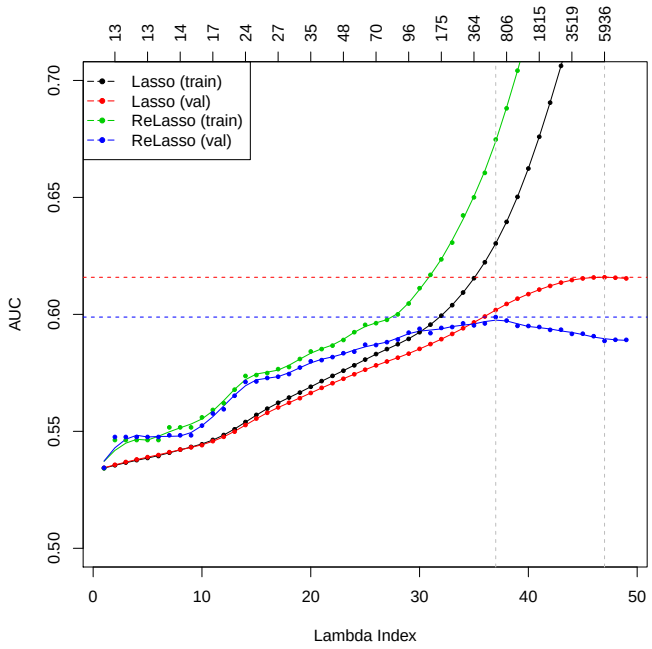


Figure 1.8: AUC plot for asthma. The top axis shows the number of active variables in the model.

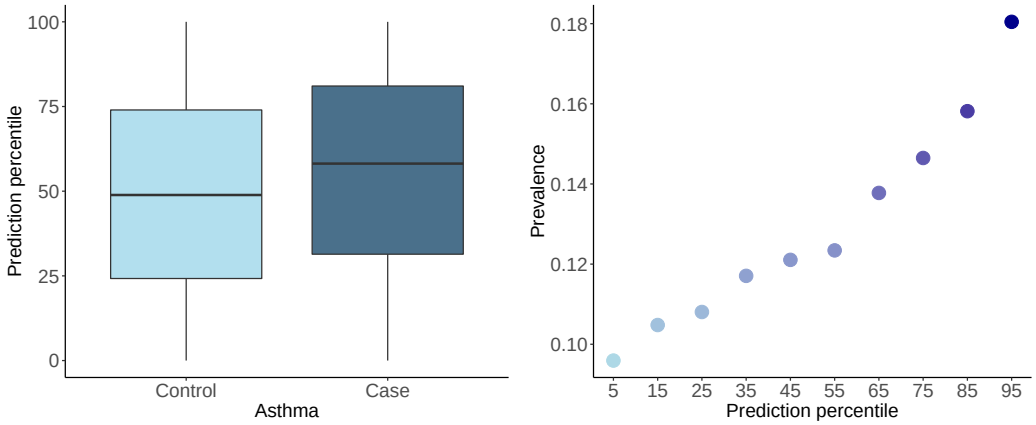


Figure 1.9: Results for asthma based on the best lasso model. Left: box plot of the percentile of the linear prediction score among cases versus controls. Right: the stratified prevalence across different percentile bins based on the predicted scores by the optimal lasso.

blood and is a risk factor for cardiovascular disease. It is highly heritable and may be polygenic. Our results are summarized in Table 1.5. The AUC curves for the lasso and the relaxed lasso are shown in Figure 1.10. Similarly the ROC curve for the best lasso model is shown in Figure 1.12,

Model	Form	AUC_{train}	AUC_{val}	AUC_{test}	Size
(1)	Age + Sex	0.6918	0.6952	0.6883	2
(2)	Age + Sex + 10 PCs	0.6927	0.6959	0.6889	12
(3)	(2) + Single SNP	0.6963	6982	0.6921	13
(4)	(2) + 10K Combined	0.7402	0.6956	0.6880	10,012
(5)	(2) + 100K Combined	0.8518	0.6607	0.6547	100,012
(6)	Sequential LR	0.7540	0.7167	0.7137	1,012
(7)	Lasso	0.7832	0.7259	**0.7191**	1,371
(8)	Relaxed Lasso	0.7273	0.7220	0.7166	239

Table 1.5: AUC values for high cholesterol. For lasso and relaxed lasso, the chosen model is based on maximum AUC on the validation set. Model (3) to (8) each includes Model (2) plus their own specification as stated in the Form column.

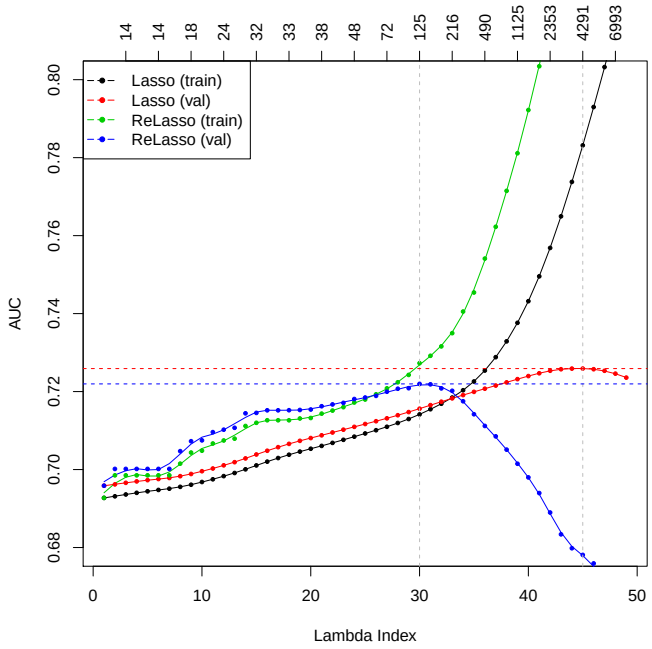


Figure 1.10: AUC plot for high cholesterol. The top axis shows the number of active variables in the model.

and box plots for the two groups and a stratified prevalence plot are shown in Figure 1.11. We see that the distributions of predictions made on non-HC individuals and on HC individuals are clearly different from each other, suggesting good classification results. That is reflected in the AUC measure listed in the table. Nevertheless, it is not much better than the result of the base model including only covariates age and sex.

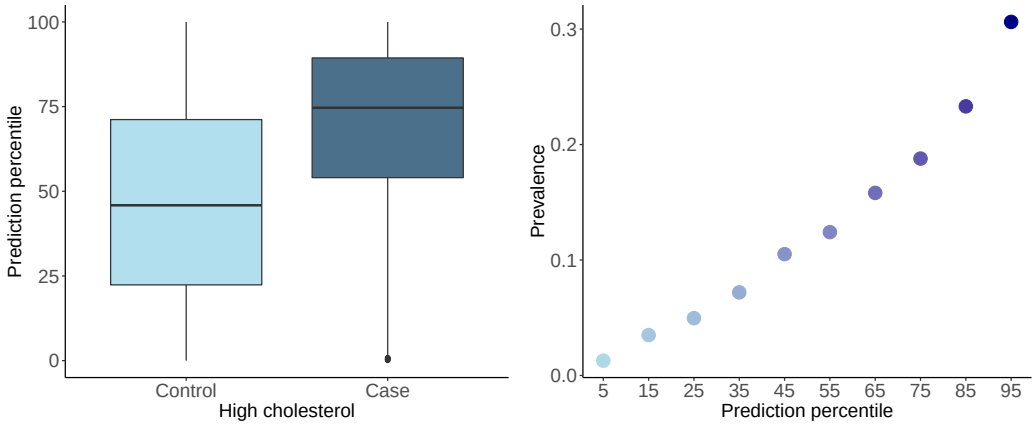


Figure 1.11: Results for high cholesterol based on the best lasso model. Left: box plot of the percentile of the linear prediction score among cases versus controls. Right: the stratified prevalence across different percentile bins based on the predicted scores by the optimal lasso.

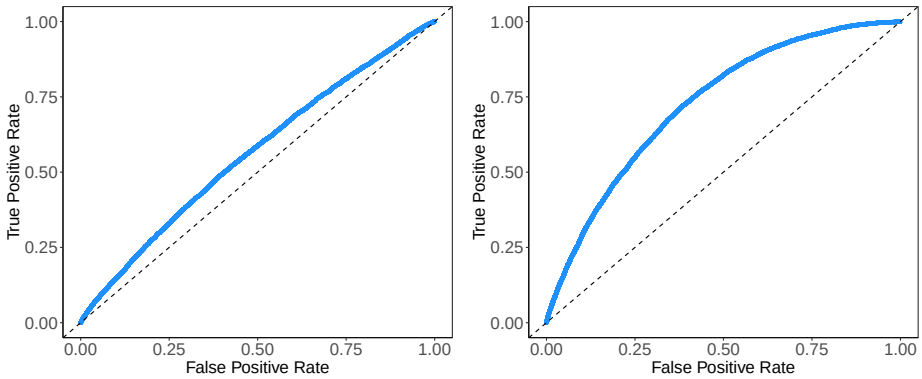


Figure 1.12: ROC curves. Left: asthma. Right: high cholesterol.

1.3 Discussion

In this paper, we propose a novel batch screening iterative lasso (BASIL) algorithm to fit the full lasso solution path for very large and high-dimensional datasets. It can be used, among the others, for Gaussian linear model, logistic regression and Cox regression, and can be easily extended to fit the elastic-net with mixed ℓ_1/ℓ_2 penalty. It enjoys the advantages of high efficiency, flexibility and easy implementation. For SNP data as in our applications, we develop an R package **snpnet** that incorporates SNP-specific optimizations and are able to process datasets of wide interest from the UK Biobank.

In our algorithm, the choice of M is important for the practical performance. It trades off

between the number of iterations and the computation per iteration. With a small M or small update of the strong set, it is very likely that we are unable to proceed fast along the λ sequence in each iteration. Although the design of the BASIL algorithm guarantees that for any $M, \Delta M > 0$, we are able to obtain the full solution path after sufficient iterations, many iterations will be needed if M is chosen too small, and the disk I/O cost will be dominant. In contrast, a large M will incur more memory burden and more expensive lasso computation, but with the hope to find more valid lasso solutions in one iteration, save the number of iterations and the disk I/O. It is hard to identify the optimal M a priori. It depends on the computing architecture, the size of the problem, the nature of the phenotype, etc. For this reason, we tend to leave it as a subjective parameter to the user's choice. However in the meantime, we do plan to provide a more systematic option to determine M, which leverages the strong rules again. Recall that in the simple setting with no intercept and no covariates, the initial strong set is constructed by $|x_j^\top y| \leq 2\lambda - \lambda_{\max}$. Since the strong rules rarely make mistakes and are fairly effective in discarding inactive variables, we can guide the choice of batch size M by the number of λ values we want to cover in the first iteration. For example, one may want the strong set to be large enough to solve for the first 10 λ's in the first iteration. We can then let $M = |\{1 \leq j \leq p : |x_j^\top y| > 2\lambda_{10} - \lambda_{\max}\}|$. Despite being adaptive to the data in some sense, this approach is by no means computationally optimal. It is more based on heuristics that the iteration should make reasonable progress along the path.

Our numerical studies demonstrate that the iterative procedure effectively reduces a big-n-big-p lasso problem into one that is manageable by in-memory computation. In each iteration, we are able to use parallel computing when applying screening rules to filter out a large number of variables. After screening, we are left with only a small subset of data on which we are able to conduct intensive computation like cyclical coordinate descent all in memory. For the subproblem, we can use existing fast procedures for small or moderate-size lasso problems. Thus, our method allows easy reuse of previous software with lightweight development effort.

When a large number of variables is needed in the optimal predictive model, it may still require either large memory or long computation time to solve the smaller subproblem. In that case, we may consider more scalable and parallelizable methods like proximal gradient descent [Parikh and Boyd, 2014] or dual averaging [Xiao, 2010, Duchi et al., 2012]. One may think why don't we directly use these methods for the original full problem? First, the ultra high dimension makes the evaluation of gradients, even on mini-batch very expensive. Second, it can take a lot more

steps for such first-order methods to converge to a good objective value. Moreover, the speed of convergence depends on the choice of other parameters such as step size and additional constants in dual averaging. For those reasons, we still prefer the tuning-free and fast coordinate descent methods when the subproblem is manageable.

The lasso has nice variable selection and prediction properties if the linear model assumption together with some additional assumptions such as the restricted eigenvalue condition [Bickel et al., 2009] or the irrepresentable condition [Zhao and Yu, 2006] holds. In practice, such assumptions do not always hold and are often hard to verify. In our UK Biobank application, we don't attempt to verify the exact conditions, and the selected model can be subject to false positives. However, we demonstrate relevance of the selection via empirical consistency with the GWAS results. We have seen superior prediction performance by the lasso as a regularized regression method compared to other methods. More importantly, by leveraging the sparsity property of the lasso, we are able to manage the ultrahigh-dimensional problem and obtain a computationally efficient solution.

When comparing with other methods in the UK Biobank experiments, due to the large number of test samples (60,000+), we are confident that the lasso and elastic-net methods are able to do significantly better than other methods. In fact, the standard error of R^2 can be easily derived by the delta method, and the standard error of the AUC can be estimated and upper bounded by $1/(4\min(m,n))$ [DeLong et al., 1988, Cortes and Mohri, 2005], where m, n represents the number of positive and negative samples. For height and BMI, it turns out that the standard errors are roughly 0.001, or 0.1%. For asthma and high cholesterol, considering the case rate around 12%, the standard errors can be upper bounded by 0.005, or 0.5%. Therefore, on height, BMI and asthma, the lasso and elastic net perform significantly better than the other methods, while on high cholesterol, the Sequential LR and the relaxed lasso have competitive performance as well.

1.4 Materials and Methods

Variants in the BASIL framework Some other very useful components can be easily incorporated into the BASIL framework. We will discuss debiasing using the relaxed lasso and the inclusion of adjustment covariates.

The lasso is known to shrink coefficients to exclude noise variables, but sometimes such shrinkage can degrade the predictive performance due to its effect on actual signal variables. Meinshausen

[2007] introduces the relaxed lasso to correct for the potential over-shrinkage of the original lasso estimator. They propose a refitting step on the active set of the lasso solution with less regularization, while a common way of using it is to fit a standard OLS on the active set. The active set coefficients are then set to

$$\hat{\beta}_{\mathcal{A},\text{Relax}}(\lambda) = \text{argmin}_{\beta_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}} \|y - X_{\mathcal{A}}\beta_{\mathcal{A}}\|_2^2,$$

whereas the coefficients for the inactive set remain at 0. This refitting step can revert some of the shrinkage bias introduced by the vanilla lasso. It doesn't always reduce prediction error due to the accompanied increase in variance when there are many variables in the model or when the signals are weak. That being said, we can still insert a relaxed lasso step with little effort in our iterative procedure: once a valid lasso solution is found for a new λ, we may refit with OLS. As we iterate, we can monitor validation error for the lasso and the relaxed lasso. The relaxed lasso will generally end up choosing a smaller set of variables than the lasso solution in the optimal model.

In some applications such as GWAS, there may be confounding variables $Z \in \mathbb{R}^{n \times q}$ that we want to adjust for in the model. Population stratification, defined as the existence of a systematic ancestry difference in the sample data, is one of the common factors in GWAS that can lead to spurious discoveries. This can be controlled for by including some leading principal components of the SNP matrix as variables in the regression [Price et al., 2006]. In the presence of such variables, we instead solve

$$(\hat{\alpha}(\lambda), \hat{\beta}(\lambda)) = \text{argmin}_{\alpha \in \mathbb{R}^q, \beta \in \mathbb{R}^p} \ \frac{1}{2n}\|y - Z\alpha - X\beta\|_2^2 + \lambda\|\beta\|_1. \tag{1.7}$$

This variation can be easily handled with small changes in the algorithm. Instead of initializing the residual with the response y, we set $r^{(0)}$ equal to the residual from the regression of y on the covariates. In the fitting step, in addition to the variables in the strong set, we include the covariates but leave their coefficients unpenalized as in (1.7). Notice that if we want to find relaxed lasso fit with the presence of adjustment covariates, we need to include those covariates in the OLS as well, i.e.,

$$(\hat{\alpha}_{\text{Relax}}(\lambda), \hat{\beta}_{\mathcal{A},\text{Relax}}(\lambda)) = \text{argmin}_{\alpha \in \mathbb{R}^q, \beta_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}} \|y - Z\alpha - X_{\mathcal{A}}\beta_{\mathcal{A}}\|_2^2. \tag{1.8}$$

UK Biobank experiment details We focused on 337,199 White British unrelated individuals out of the full set of over 500,000 from the UK Biobank dataset [Bycroft et al., 2018] that satisfy the same set of population stratification criteria as in DeBoever et al. [2018]: (1) self-reported White British ancestry, (2) used to compute principal components, (3) not marked as outliers for heterozygosity and missing rates, (4) do not show putative sex chromosome aneuploidy, and (5) have at most 10 putative third-degree relatives. These criteria are meant to reduce the effect of confoundedness and unreliable observations.

The number of samples is large in the UK Biobank dataset, so we can afford to set aside an independent validation set without resorting to the costly cross-validation to find an optimal regularization parameter. We also leave out a subset of observations as test set to evaluate the final model. In particular, we randomly partition the original dataset so that 60% is used for training, 20% for validation and 20% for test. The lasso solution path is fit on the training set, whereas the desired regularization is selected on the validation set, and the resulting model is evaluated on the test set.

We are going to further discuss some details in our application that one might also encounter in practice. They include adjustment for confounders, missing value imputation and variable standardization in the algorithm.

In genetic studies, spurious associations are often found due to confounding factors. Among the others, one major source is the so-called population stratification [Patterson et al., 2006]. To adjust for that effect, it is common is to introduce the top principal components and include them in the regression model. Therefore in the lasso method, we are going to solve (1.7) where in addition to the SNP matrix X, we let Z include covariates such as age, sex and the top 10 PCs of the SNP matrix.

Missing values are present in the dataset. As quality control normally done in genetics, we first discard observations whose phenotypic value of interest is not available. We further exclude variants whose missing rate is greater than 10% or the minor allele frequency (MAF) is less than 0.1%, which results in around 685,000 SNPs for height. In particulr, 685,362 for height, 685,371 for BMI, 685,357 for asthma and 685,357 for HC. The number varies because the criteria are evaluated on the subset of individuals whose phenotypic value is observed (after excluding the missing ones), which can be different across different phenotypes. For those remaining variants, mean imputation is conducted to fill the missing SNP values; that is, the missing values in every SNP are imputed

with the mean observed level of that SNP in the population under study.

When it comes to the lasso fitting, there are some subtleties that can affect its variable selection and prediction performance. One of them is variable standardization. It is often a step done without much thought to deal with heterogeneity in variables so that they are treated fairly in the objective. However in our studies, standardization may create some undesired effect. To see this, notice that all the SNPs can only take values in 0, 1, 2 and NA — they are already on the same scale by nature. As we know, standardization would use the current standard deviation of each predictor as the divisor to equalize the variance across all predictors in the lasso fitting that follows. In this case, standardization would unintentionally inflate the magnitude of rare variants and give them an advantage in the selection process since their coefficients effectively receive less penalty after standardization. In Figure 1.13, we can see the distribution of standard deviation across all variants in our dataset. Hence, to avoid potential spurious findings, we choose not to standardize the variants in the experiments.

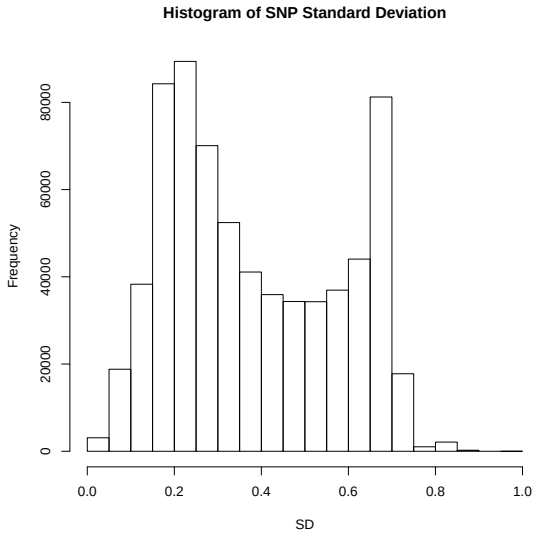


Figure 1.13: Histogram of the standard deviations of the SNPs. They are computed *after* mean imputation of the missing values because they would be the exact standardization factors to be used if the lasso were applied with variable standardization on the mean-imputed SNP matrix.

Computational optimization in software implementation Among the iterative steps in BASIL, screening and checking are where we need to deal with the full dataset. To deal with the

memory bound, we can use memory-mapped I/O. In R, **bigmemory** [Kane et al., 2013] provides a convenient implementation for that purpose. That being said, we do not want to rely on that for intensive computation modules such as cyclic coordinate descent, because frequent visits to the on-disk data would still be slow. Instead, since the subset of strong variables would be small, we can afford to bring them to memory and do fast lasso fitting there. We only use the full memory-mapped dataset in KKT checking and screening. Moreover since checking in the current iteration can be done together with the screening in the next iteration, effectively only one expensive pass over the full dataset is needed every iteration.

In addition, we use a set of techniques to speed up the computation. First, the KKT check can be easily parallelized by splitting on the features when multi-core machines are available. The speedup of this part is immediate and (slightly less than) proportional to the number of cores available. Second, specific to the application, we exploit the fact that there are only 4 levels for each SNP value and design a faster inner product routine to replace normal float number multiplication in the KKT check step. In fact, given any SNP vector $x \in \{0, 1, 2, \mu\}^n$ where μ is the imputed value for the missing ones, we can write the dot product with a vector $r \in \mathbb{R}^n$ as

$$x^\top r = \sum_{i=1}^{n} x_i r_i = 1 \cdot \sum_{i:x_i=1} r_i + 2 \cdot \sum_{i:x_i=2} r_i + \mu \cdot \sum_{i:x_i=\mu} r_i.$$

We see that the terms corresponding to 0 SNP value can be ignored because they don't contribute to the final result. This will significantly reduce the number of arithmetic operations needed to compute the inner product with rare variants. Further, we only need to set up 3 registers, each for one SNP value accumulating the corresponding terms in r. A series of multiplications is then converted to summations. In our UK Biobank studies, although the SNP matrix is not sparse enough to exploit sparse matrix representation, it still has around 70% 0's. We conduct a small experiment to compare the time needed to compute $X^\top R$, where $X \in \{0, 1, 2, 3\}^{n \times p}, R \in \mathbb{R}^{p \times k}$. The proportions for the levels in X are about $70\%, 10\%, 10\%, 10\%$, similar to the distribution of SNP levels in our study, and R resembles the residual matrix when checking the KKT condition. The number of residual vectors is $k = 20$. The mean time over 100 repetitions is shown in Table 1.6.

We implement the procedure with all the optimizations in an R package called **snpnet**, which is currently available at `https://github.com/junyangq/snpnet`. It assumes pgen file format [Chang

Multiplication Method	$n = 200, p = 800$	$n = 2000, p = 8000$
Standard	3.20	306.01
SNP-Optimized	1.32	130.21

Table 1.6: Timing performance (milliseconds) on multiplication of SNP matrix and residual matrix. The methods are all implemented in C++ and run on a Macbook with 2.9 GHz Intel Core i7 and 8 GB 1600 MHz DDR3.

et al., 2015] of the SNP matrix, fits the lasso solution path and allows early stopping if a validation dataset is provided. In order to achieve better efficiency, we suggest using **snpnet** together with **glmnetPlus**, a warm-started version of **glmnet**, which is currently available at `https://github.com/junyangq/glmnetPlus`. It allows one to provide a good initialization of the coefficients to fit part of the solution path instead of always starting from the all-zero solution by **glmnet**.

Related methods and packages There are a number of existing screening rules for solving big lasso problems. Sobel et al. [2009] use a screened set to scale down the logistic lasso problem and check the KKT condition to validate the solution. Their focus, however, is on selecting a lasso model of particular size and only the initial screened set is expanded if the KKT condition is violated. In contrast, we are interested in finding the whole solution path (before overfitting). We adopt a sequential approach and keep updating the screened set at each iteration. This allows us to potentially keep the screened set small as we move along the solution path. Other rules include the SAFE rule [El Ghaoui et al., 2010], Sure Independence Screening [Fan and Lv, 2008], and the DPP and EDPP rules [Wang et al., 2015].

We expand the discussion on these screening rules a bit. Fan and Lv [2008] exploits marginal information of correlation to conduct screening but the focus there is not optimization algorithm. Most of the screening rules mentioned above (except for EDPP) use inner product with the current residual vector to measure the importance of each predictor at the next λ — those under a threshold can be ignored. The key difference across those rules is the threshold defined and whether the resulting discard is safe. If it is safe, one can guarantee that only one iteration is needed for each λ value, compared with others that would need more rounds if an active variable was falsely discarded. Though the strong rules rarely make this mistake, safe screening is still a nice feature to have in single-λ solutions. However, under the batch mode we consider due to the desire of reducing the number of full passes over the dataset, the advantage of safe threshold may not be as much. In fact, one way we might be able to leverage the safe rules in the batch mode is to first find out the

set of candidate predictors for the several λ values up to λ_k we wish to solve in the next iteration based on the current inner products and the rules' safe threshold, and then solve the lasso for these parameters. Since these rules can often be conservative, we would then have strong incentive to solve for, say, one further λ value λ_{k+1} because if the current screening turns out to be a valid one as well, we will find one more lasso solution and move one step forward along the λ sequence we want to solve for. This can potentially save one iteration of the procedure and thus one expensive pass over the dataset. The only cost there is computing the lasso solution for one more λ_{k+1} and computing inner products with one more residual vector at λ_{k+1} (to check the KKT condition). The latter can be done in the same pass as we compute inner products at λ_k for preparing the screening in the next iteration, and so no additional pass is needed. Thus under the batch mode, the property of safe screening may not be as important due to the incentive of aggressive model fitting. Nevertheless it would be interesting to see in the future EDPP-type batch screening. It uses inner products with a modification of the residual vector. Our algorithm still focuses of inner products with the vanilla residual vector.

To address the large-scale lasso problems, several packages have been developed such as **biglasso** [Zeng and Breheny, 2017], **bigstatsr** [Privé et al., 2018], **oem** [Huling and Qian, 2018] and the lasso routine from **PLINK** 1.9 [Chang et al., 2015].

Among them, **oem** specializes in tall data (big n) and can be slow when $p > n$. In many real data applications including ours, the data are both large-sample and high-dimensional. However, we might still be able to use **oem** for the small lasso subroutine since a large number of variables have already been excluded. The other packages, **biglasso**, **bigstatsr**, **PLINK** 1.9, all provide efficient implementations of the pathwise coordinate descent with warm start. **PLINK** 1.9 is specifically developed for genetic datasets and is widely used in GWAS and research in population genetics. In **bigstatsr**, the `big_spLinReg` function adapts from the `biglasso` function in **biglasso** and incorporates a Cross-Model Selection and Averaging (CMSA) procedure, which is a variant of cross-validation that saves computation by directly averaging the results from different folds instead of retraining the model at the chosen optimal parameter. They both use memory-mapping to process larger-than-RAM, on-disk datasets as if they were in memory, and based on that implement coordinate descent with strong rules and warm start.

The main difference between BASIL and the algorithm these packages use is that BASIL tries to solve a series of models every full scan of the dataset (at checking and screening) and thus effectively

reduce the number of passes over the dataset. This difference may not be significant in small or moderate-sized problems, but can be critical in big data applications especially when the dataset cannot be fully loaded into the memory. A full scan of a larger-than-RAM dataset can incur a lot of swap-in/out between the memory and the disk, and thus a lot of disk I/O operations, which is known to be orders of magnitude slower than in-memory operations. Thus reducing the number of full scans can greatly improve the overall performance of the algorithm.

Aside from potential efficiency consideration, all of those packages aforementioned have to re-implement a variety of features existent in many small-data solutions but for big-data context. Nevertheless, currently they don't provide as much functionality as needed in our real-data application. First, in the current implementations, **PLINK** 1.9 only supports the Gaussian family, **biglasso** and **bigstatsr** only supports the Gaussian and binomial families, whereas **snpnet** can easily extend to other regression families and already built in Gaussian, binomial and Cox families. Also, **biglasso**, **bigstatsr** and **PLINK** 1.9 all standardize the predictors beforehand, but in many applications such as our UK Biobank studies, it is more reasonable to leave the predictors unstandardized. In addition, it can take some effort to convert the data to the desired format by these packages. This would be a headache if the raw data is in some special format and one cannot afford to first convert the full dataset into an intermediate format for which a tool is provided to convert to the desired one by **biglasso** or **bigstatsr**. This can happen, for example, if the raw data is highly compressed in a special format. For the BED binary format we work with in our application, `readRAW_big.matrix` function from **BGData** can convert a raw file to a `big.matrix` object desired by **biglasso**, and `snp_readBed` function from **bigsnpr** [Privé et al., 2018] allows one to convert it to `FBM` object desired by **bigstatsr**. However, **bigsnpr** doesn't take input data that has any missing values, which can prevalent in an SNP matrix (as in our application). Although **PLINK** 1.9 works directly with the BED binary file, its lasso solver currently only supports the Gaussian family, and it doesn't return the full solution path. Instead it returns the solution at the smallest λ value computed and needs a good heritability estimate as input from the user, which may not be immediately available.

We summarize the main advantages of the BASIL algorithm:

- **Input data flexibility**. Our algorithm allows one to deal directly with any data type as long as the screening and checking steps are implemented, which is often very lightweight development work like matrix multiplication. This can be important in large-scale applications

especially when the data is stored in a compressed format or a distributed way since then we would not need to unpack the full data and can conduct KKT check and screening on its original format. Instead only a small screened subset of the data needs to be converted to the desired format by the lasso solver in the fitting step.

- **Model flexibility**. We can easily transfer the modeling flexibility provided by existing packages to the big data context, such as the options of standardization, sample weights, lower/upper coefficient limits and other families in generalized linear models provided by existing packages such as **glmnet**. This can be useful, for example, when we may not want to standardize predictors already in the same unit to avoid unintentionally different penalization of the predictors due to difference in their variance.

- **Effortless development**. The BASIL algorithm allows one to maximally reuse the existing lasso solutions for small or moderate-sized problems. The main extra work would be an implementation of batch screening and KKT check with respect to a particular data type. For example, in the **snpnet** package, we are able to quickly extend the in-memory **glmnet** solution to large-scale, ultrahigh-dimentional SNP data. Moreover, the existing convenient data interface provided by the **BEDMatrix** package further facilitates our implementation.

- **Computational efficiency**. Our design reduces the number of visits to the original data that sits on the disk, which is crucial to the overall efficiency as disk read can be orders of magnitude slower than reading from the RAM. The key to achieving this is to bring batches of promising variables into the main memory, hoping to find the lasso solutions for more than one λ value each iteration and check the KKT condition for those λ values in one pass of the entire dataset.

Lastly, we are going to provide some timing comparison with existing packages. As mentioned in previous sections, those packages provide different functionalities and have different restrictions on the dataset. For example, most of them (**biglasso, bigstatsr**) assume that there are no missing values, or the missing ones have already been imputed. In **bigsnpr**, for example, we shouldn't have SNPs with 0 MAF either. Some packages always standardize the variants before fitting the lasso. To provide a common playground, we create a synthetic dataset with no missing values, and follow a standardized lasso procedure in the fitting stage, simply to test the computation. The dataset has

R Package	Elapsed Time (minutes)
bigstatsr [Privé et al., 2018]	2.93 + 56.80
bigstatsr + CMSA [Privé et al., 2018]	2.93 + 101.75
biglasso[Zeng and Breheny, 2017]	4.55 + 54.27
PLINK [Chang et al., 2015]	53.52
snpnet	**44.79**

Table 1.7: Timing comparison on a synthetic dataset of size $n = 50,000$ and $p = 100,000$. The time for **bigstatsr** and **biglasso** has two components: one for the conversion to the desired data type and the other for the actual computation. The experiments are all run with 16 cores and 64 GB memory.

50,000 samples and 100,000 variables, and each takes value in the SNP range, i.e., in 0, 1, or 2. We fit the first 50 lasso solutions along a prefix λ sequence that contains 100 initial λ values (like early stopping for most phenotypes). The total time spent is displayed in Table 1.7. For **bigstatsr**, we include two versions since it does cross-validation by default. In one version, we make it comply with our single train/val/test split, while in the other version, we use its default 10-fold cross-validation version — Cross-Model Selection and Averaging (CMSA). Notice that the final solution of iCMSA is different from the exact lasso solution on the full data because the returned coefficient vector is a linear combination of the coefficient vectors from the 10 folds rather than from a retrained model on the full data. We uses 128GB memory and 16 cores for the computation.

From the table, we see that **snpnet** is at about 20% faster than other packages concerned. The numbers before the "+" sign are the time spent on converting the raw data to the required data format by those packages. The second numbers are time spent on actual computation.

It is important to note though that the performance relies not only on the algorithm, but also heavily on the implementations. The other packages in comparison all have their major computation done with C++ or Fortran. Ours, for the purpose of meta algorithm where users can easily integrate with any lasso solver in R, still has a significant portion (the iterations) in R and multiple rounds of cross-language communication. That can degrade the timing performance to some degree. If there is further pursuit of speed performance, there is still space for improvement by more designated implementation.

1.5 Appendix: Manhattan Plots

The Manhattan plots in Figure 1.14 (generated using the **qqman** package [Turner, 2018]) show the magnitude of the univariate p-values and the size of the lasso coefficients for each gene for the two quantitative traits and two binary traits. The coefficients are plotted for the model with the optimal R^2 value on the validation set. The variants highlighted in green in both plots are those that have coefficient magnitudes above the 99th percentile of all coefficient magnitudes for the trait. The horizontal line in the p-value plot is plotted at the genome-wide Bonferroni corrected p-value threshold 5×10^{-8}. There are two main points we would like to highlight:

- The lasso manages to capture significant univariate predictors in each genetic region. Due to possible correlation it does not pick up the variants with similarly small p-values located nearby.
- Some of the variants with weak univariate signals are also identified and turn out to be crucial to the predictive performance of the lasso.

For the two qualitative traits plotted in Figure 1.15, there are fewer p-values above the threshold, and many of the significant ones are located close to each other. The size of the lasso fit is correspondingly smaller, and the large coefficients pick up the important locations as before. However, the nonzero coefficients are still spread across the whole genome.

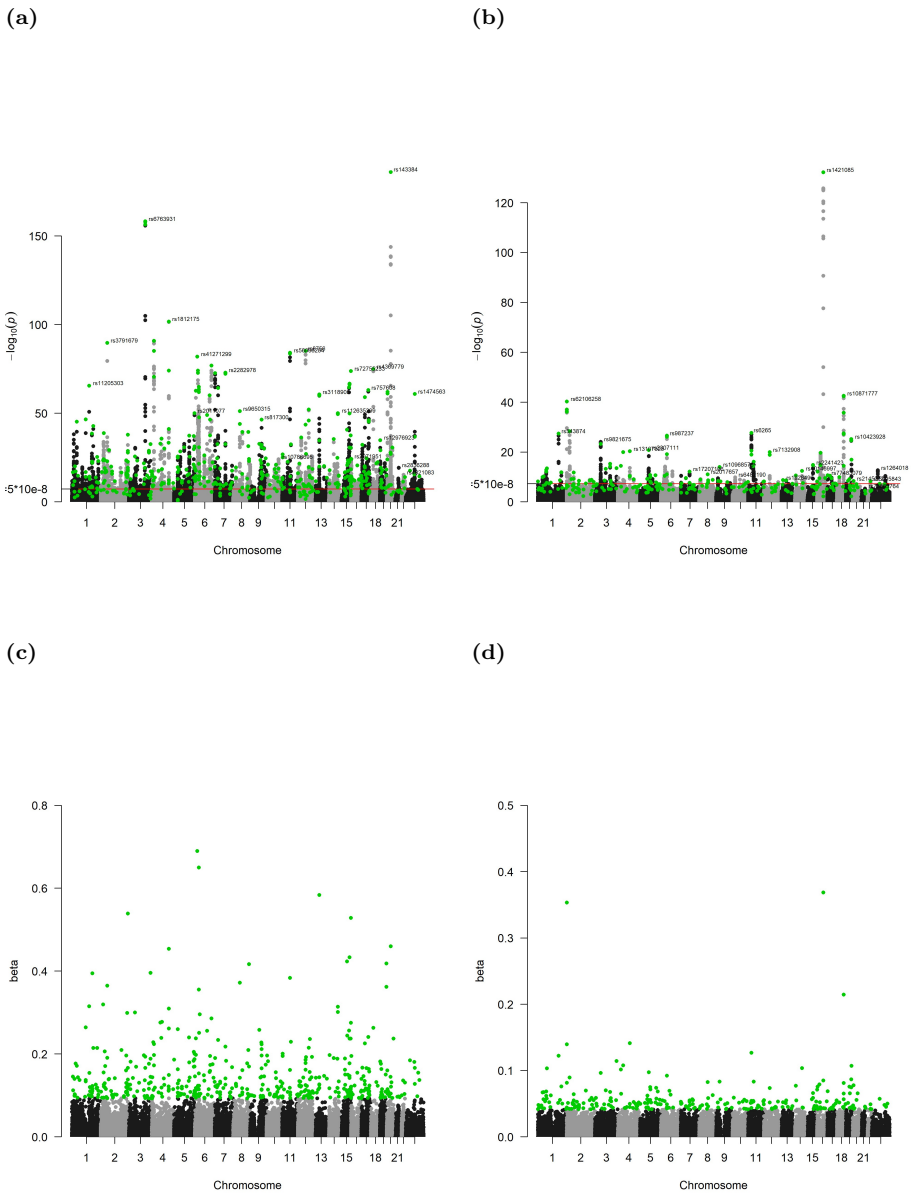


Figure 1.14: Manhattan plots of the univariate p-values and lasso coefficients for height (a, c) and BMI (b, d). The vertical axis of the p-value plots shows $-\log_{10}(p)$ for each SNP, while the vertical axis of the coefficient plots shows the magnitude of the coefficients from **snpnet**. The SNPs with relatively large lasso coefficients are highlighted in green. The red horizontal line on the p-value plot represents a reference level of $p = 5 \times 10^{-8}$.

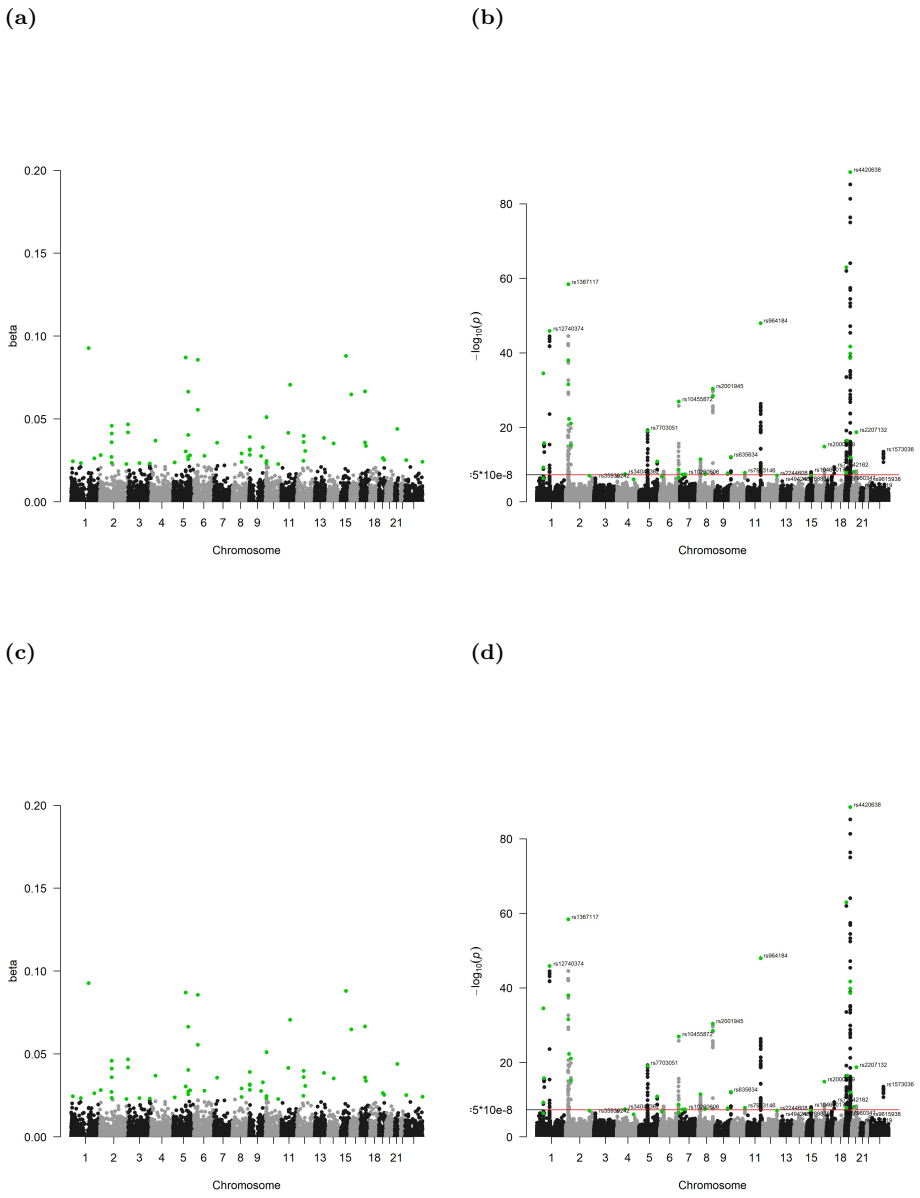


Figure 1.15: Manhattan plots of the univariate p-values and lasso coefficients for asthma (a, c) and high cholesterol (b, d). The vertical axis of the p-value plots shows $-\log_{10}(p)$ for each SNP, while the vertical axis of the coefficient plots shows the magnitude of the coefficients from **snpnet**. The SNPs with relatively large lasso coefficients are highlighted in green. The red horizontal line on the p-value plot represents a reference level of $p = 5 \times 10^{-8}$.

Chapter 2

Fast Reduced Rank Regression for Large-scale and Ultrahigh-dimensional Problems

2.1 Introduction

The past two decades have witnessed rapid growth in the amount of data available to us. Many areas such as genomics, neuroscience, economics and Internet services have been producing increasingly larger datasets that have high dimension, large sample size, or both. A variety of statistical methods and computational tools have been developed to accommodate this change so that we are able to extract valuable information and insight from these massive datasets [Friedman et al., 2009, Efron and Hastie, 2016, Dean and Ghemawat, 2008, Zaharia et al., 2010, Abadi et al., 2016].

One major motivating application for this work is the study of data from population-scale cohorts like UK Biobank with genetic data from over one million genetic variants and phenotype data from thousands of phenotypes in over 500,000 individuals [Bycroft et al., 2018]. These data present unprecedented opportunities to explore very comprehensive genetic relationships with phenotypes of interest. In particular, the subset of tasks we are interested in is the prediction of a person's phenotype value, such as disease affection status, based on his or her genetic variants.

Genome-wide association studies (GWAS) is a very powerful and widely used framework for identifying genetic variants that are associated with a given phenotype. See, for example, Visscher et al. [2017] and the references therein. It is based on the results of univariate marginal regression over all candidate variants and tries to find a subset of significant ones. While being computationally efficient and easy to interpret, GWAS has fairly limited prediction performance because at most one predictor can present in the model. If prediction performance is our main concern, it is natural to consider the class of multivariate methods, i.e. that which considers multiple variants simultaneously. In the past, *wide* data were prevalent where only a limited number, like thousands, of samples were available. In this regime, some sophisticated multivariate methods could be applicable, though they have to more or less deal with dimension reduction or variable selection. In this setting, we handle hundreds of thousands samples and even more variables. In such cases, statistical methods and computational algorithms become equally important because only efficient algorithmic design will allow for the application of sophisticated statistical modeling. Recently, we introduced some algorithms addressing these challenges. In particular, Qian et al. [2019] proposed an iterative screening framework that is able to fit the exact lasso/elastic-net solution path in large-scale and ultrahigh-dimensional settings, and demonstrate competitive computational efficiency and superior prediction performance over previous methods.

In this paper, we consider the scenarios where multivariate responses are available in addition to the multiple predictors, and propose a suite of statistical methods and efficient algorithms that allow us to further improve the statistical performance in this large n and large p regime. Some characteristics we want to leverage and challenges we want to solve include:

Statistics There are thousands of phenotypes available in the UK Biobank. Many of them are highly correlated with each other and can have a lot of overlap in their driving factors. By treating them separately, we lose this information that could have been used to stabilize our model estimation. The benefit of building a joint model can be seen from the following simplified model. Suppose all the outcomes $\mathbf{y}^k, k = 1, \ldots, q$ are independent noisy observations of a shared factor $\mathbf{u} = \mathbf{X}\beta$ such that $\mathbf{y}^k = \mathbf{u} + \mathbf{e}^k$. It is easy to see that by taking an average across all the outcomes, we obtain a less noisy response $\bar{\mathbf{y}}$, and this will give us more accurate parameter estimation and better prediction than the model built on any of the single outcome. The assumption of such latent structure is an important approach to capturing the correlation structure among the outcomes and

can bring in a significant reduction in variance if the data indeed behave in a similar way. We will formalize this belief and build a model on top of it. In addition, in the presence of high-dimensional features, we will follow the "bet on sparsity" principle [Friedman et al., 2009], and assume that only a subset of the predictors are relevant to the prediction.

Therefore, the statistical model we will build features two major assumptions: **low-rank** in the signal and **sparse effect**. Furthermore, we will introduce integrated steps to systematically deal with confounders and missing values.

Computation On a large-scale dataset, building a multivariate model can pose great computational challenge. For example, loading the entire UK Biobank dataset into memory with double precision will take more than one terabyte of space, while the foremost assumption of most existing statistical computing tools assume that the data are already sitting in memory. Even if large memory is available, one can always encounter data or construct features so that it becomes insufficient. Hence, instead of expecting sufficient memory space, we would like to find a scalable solution that is less restricted to the size of physical memory.

There is a dynamic data access mechanism provided by the operating system called memory mapping [Bovet and Cesati, 2005] that allows for easy access to larger-than-memory data on the disk. In essence, it carries a chunk of data from disk to memory when needed and swap some old chunks of data out of memory when it is full. In principle, we could add a layer of memory mapping on top of all the procedures and then access the data as if they were in memory. However, there is one important practical component that should never be ignored: disk I/O. This is known to be expensive in the operating system and can greatly delay the computation if frequent disk I/Os are involved. For this reason, we do not pursue first-order gradient-based methods such as stochastic gradient descent [Bottou, 2010] or dual averaging [Xiao, 2010, Duchi et al., 2011] because it can take a large number of passes over the data for the objective function to converge to the optimum.

To address this, we design the algorithm so that it needs as few full passes over the data as possible while solving the exact objective. In particular, by leveraging the sparsity assumption, we propose an adaptive screening approach that allows us to strategically select a small subset of variables into memory, do intensive computation on the subset, and then verify the validity of all the left-out variables. The last step is important because we want to guarantee that the solution obtained from the algorithm is a valid solution to the original full problem.

2.1.1 Reduced-Rank Regression for Multiple Responses

In the standard multivariate linear regression model, given a model matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p) \in \mathbb{R}^{n \times p}$ and a multivariate response matrix $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_q) \in \mathbb{R}^{n \times q}$, we assume that

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E},$$

where each row of $\mathbf{E} = (\mathbf{e}_1, \ldots, \mathbf{e}_q)$ is assumed to be an independent sample from some multivariate Gaussian distribution $\mathbf{E}^{(i)} \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma_E)$. When $n \geq q$, it is easy to see that an maximum likelihood estimator (MLE) can be found by solving a least squares problem with multiple outcomes, i.e.

$$\hat{\mathbf{B}} \in \operatorname{argmin}_{\mathbf{B} \in \mathbb{R}^{p \times q}} \frac{1}{2} \|\mathbf{Y} - \mathbf{XB}\|_F^2,$$

where $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{A}_{ij}^2}$ is the Frobenius norm of a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$. When $n \geq p$ and $\mathbf{X}$ has full rank, it has the closed-form solution $\hat{\mathbf{B}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$. Notice that this is equivalent to solving q single-response regression problems separately.

However, in many scenarios, there can be some correlation structure in the signals that we can capture to improve the statistical efficiency of the estimator. One approach to modeling the correlation is to assume that there is a set of latent factors that act as the drivers for all the outcomes. When we assume that the dependencies of the latent factors on the raw features and the outcomes on the latent factors are both linear, it is equivalent to making a low-rank assumption on the coefficient matrix. Named by Anderson et al. [1951], a reduced-rank regression (RRR) assumes that the coefficient matrix $\mathbf{B}$ has a fixed rank $r \leq \min(p, q)$, or

$$\mathbf{B} = \mathbf{UV}^\top,$$

where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_r) \in \mathbb{R}^{p \times r}, \mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_q)^\top \in \mathbb{R}^{q \times r}$.[1] With the decomposed coefficient matrices, an alternative way to express the multivariate model is to assume that there exists a set

[1] We use $\mathbf{v}_k^\top$ to represent the kth row of $\mathbf{V}$ for convenience.

of latent factors $\{\mathbf{z}_\ell \in \mathbb{R}^n : 1 \le \ell \le r\}$ such that for each ℓ,

$$\begin{aligned} \mathbf{z}_\ell &= \mathbf{X}\mathbf{u}_\ell, \\ \mathbf{y}_k &= \mathbf{Z}\mathbf{v}_k + \mathbf{e}_k. \end{aligned}$$

Figure 2.1 gives a visualization of the dependency structure described above. We notice that under the decomposition, the parameters are not identifiable. In fact, if we apply any nonsingular linear transformation $\mathbf{M} \in \mathbb{R}^{r\times r}$ such that $\mathbf{V}' = \mathbf{V}\mathbf{M}^\top$ and $\mathbf{U}' = \mathbf{U}\mathbf{M}^{-1}$, it yields the same model but different parameters. As a result, we also have an infinite number of MLEs.

Under the rank constraint, an explicit global solution can be obtained. Let $\mathbf{M}\mathbf{D}\mathbf{N}^\top$ be the singular value decomposition (SVD) of $(\mathbf{X}^\top\mathbf{X})^{-\frac{1}{2}}\mathbf{X}^\top\mathbf{Y}$, a set of solution is given by $\hat{\mathbf{U}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}\mathbf{N}$, $\hat{\mathbf{V}} = \mathbf{N}$. Velu and Reinsel [2013] has a comprehensive discussion on the model under classical large n settings.

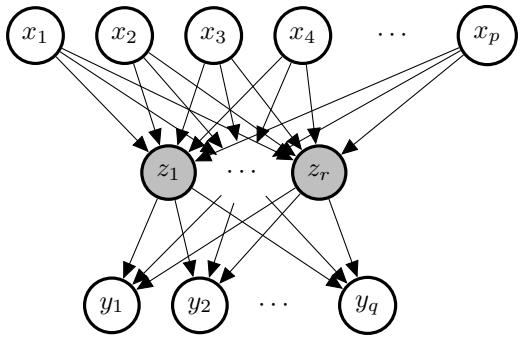


Figure 2.1: Diagram of the reduced rank regression. The nodes in grey are latent variables. The arrows represent the dependency structure.

2.1.2 Sparse Models in High-Dimensional Problems

In the setting of high-dimensional problems where $p > n$, the original low-rank coefficient matrix $\mathbf{B}$ can be unidentifiable. Often sparsity is assumed in the coefficients to model the belief that only a subset of the features are relevant to the outcomes. To find such a sparse estimate of the coefficients, a widely used approach is to add an appropriate non-smooth penalty to the original objective function to encourage the desired sparsity structure. Common choices include the lasso penalty [Tibshirani, 1996], the elastic-net penalty [Zou and Hastie, 2005] or the group lasso penalty [Yuan

and Lin, 2006]. There has been a great amount of work studying the consistency of estimation and model selection under such settings. See Greenshtein and Ritov [2004], Meinshausen and Bühlmann [2006], Zhao and Yu [2006], Bach [2008], Wainwright [2009], Bickel et al. [2009], Obozinski et al. [2011], Bühlmann and Van De Geer [2011] and references therein. In particular, the group lasso, as the name suggests, encourages group-level sparsity induced by the following penalty term:

$$P_g(\beta) = \sum_{j=1}^{J} \|\beta_j\|_2,$$

where $\beta_j \in \mathbb{R}^{p_j}$ is the subvector corresponding the jth group of variables and $\|\beta_j\|_2 = \sqrt{\sum_{\ell=1}^{p_j} \beta_{j,\ell}^2}$ is the vector ℓ_2-norm. The ℓ_2-norm enforces that if the fitted model has $\|\hat{\beta}_j\|_2 = 0$, all the elements in $\hat{\beta}_j$ will be 0, and otherwise with probability one all the elements will be nonzero. This yields a desired group-level selection in many applications. Throughout the paper, we will adopt the group lasso penalty, defining each predictor's coefficients across all outcomes as a distinct group, in order to achieve homogeneous sparsity across multiple outcomes. In addition to variable selection for better prediction and interpretation, we will also see the computational advantages we leverage to develop an efficient algorithm.

2.2 Sparse Reduced-Rank Regression

Given a rank r, we are going to solve the following penalized bounded-rank optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{Y} - \mathbf{XB}\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{B}_{j\cdot}\|_2, \\ \text{s.t.} \quad & \text{rank}(\mathbf{B}) \leq r. \end{aligned} \tag{2.1}$$

Alternatively, we can decompose the matrix explicitly as $\mathbf{B} = \mathbf{UV}^\top$ where $\mathbf{U} \in \mathbb{R}^{p\times r}, \mathbf{V} \in \mathbb{R}^{q\times r}$. It can be shown that the problem above is equivalent to the Sparse Reduced Rank Regression (SRRR) proposed by Chen and Huang [2012]:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{Y} - \mathbf{XUV}^\top\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2, \\ \text{s.t.} \quad & \mathbf{V}^\top\mathbf{V} = \mathbf{I}. \end{aligned} \tag{2.2}$$

Alternating minimization was proposed by Chen and Huang [2012] to solve this non-convex optimization problem, where two algorithms — subgradient descent and a variational method — were considered. The subgradient method was showed to be faster when $p \gg n$ and the variational method faster when $n \gg p$. However, in each iteration, the computational complexity of either method is at least quadratic in the number of variables p. It makes the problem almost intractable in ultrahigh-dimensional problems, which is common, for example, in modern genetic studies. Moreover, to obtain a model with good prediction performance, we are interested in solving the problem over multiple λ's rather than a single one. For such purposes, we design a path algorithm with adaptive variable screening that will be both memory and computationally efficient.

2.3 Fast Algorithms for Large-Scale and Ultrahigh-Dimensional Problems

First, we present a naive version of the path solution, which will be the base of our subsequent development. The path is defined on a decreasing sequence of λ values $\lambda_{\max} = \lambda_1 > \lambda_2 > \cdots > \lambda_L \geq 0$, where $\lambda_{\max}$ is often defined by one that leads to the trivial (e.g. all zero) solution and the rest are often determined by an equally spaced array on the log scale. In particular, for Problem (2.2), we are able to figure out the exact lower bound of $\lambda_{\max}$ for which the solution is trivial.

Lemma 1. In problem (2.1), if $r > 0$, the maximum λ that results in a nontrivial solution $\hat{B}(\lambda)$ is

$$\lambda_{\max} = \max_{1 \leq j \leq p} \|\mathbf{x}_j^\top \mathbf{Y}\|_2.$$

The proof is straightforward, which is a result of the Karush–Kuhn–Tucker (KKT) condition (See Boyd and Vandenberghe [2004] for more details). We present the full argument in Appendix 2.9.1. The naive path algorithm tries to solve the problem independently across different λ values.

2.3.1 Alternating Minimization

The algorithm is described in Algorithm 2. For each λ value, it applies alternating minimization to Problem (2.2) till convergence.

In the V-step (2.3), we will be solving the orthogonal Procrustes problem given a fixed $\mathbf{U}^{(k)}$.

Algorithm 2 Alternating Minimization

Define a sequence of λ values $\lambda_1 > \cdots > \lambda_L \geq 0$.
for $\ell = 1$ **to** L **do**
  Let $k = 0$, and initialize $\mathbf{U}^{(0)}, \mathbf{V}^{(0)}$.
  while $k = 0$ **or** $\|\mathbf{U}^{(k)}{\mathbf{V}^{(k)}}^\top - \mathbf{U}^{(k-1)}{\mathbf{V}^{(k-1)}}^\top\| > \epsilon$ **do**
    V-step: Fix $\mathbf{U}^{(k)}$, solve $\mathbf{V}$: the orthogonal Procrustes problem

$$\underset{\mathbf{V}:\mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\text{minimize}} \ \|\mathbf{Y} - \mathbf{X}\mathbf{U}^{(k)}\mathbf{V}^\top\|_F^2. \tag{2.3}$$

    Let $\mathbf{Y}^\top\mathbf{X}\mathbf{U}^{(k)} = \mathbf{M}\mathbf{D}\mathbf{N}^\top$ (skinny SVD) and solve $\mathbf{V}^{(k+1)} = \mathbf{M}\mathbf{N}^\top$.
    U-step: Fix $\mathbf{V}^{(k+1)}$, solve $\mathbf{U}$: the group lasso problem

$$\underset{\mathbf{U}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{Y}\mathbf{V}^{(k+1)} - \mathbf{X}\mathbf{U}\|_F^2 + \lambda_\ell \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2. \tag{2.4}$$

    $k = k + 1$
  end while
end for

An explicit solution can be constructed from the singular value decomposition, as detailed in the following Lemma.

Lemma 2. Suppose $p \geq r$ and $\mathbf{Z} \in \mathbb{R}^{p\times r}$. Let $\mathbf{Z} = \mathbf{M}\mathbf{D}\mathbf{N}^\top$ be its (skinny) singular value decomposition, where $\mathbf{M} \in \mathbb{R}^{p\times r}, \mathbf{D} = \mathbb{R}^{r\times r}$ and $\mathbf{N} \in \mathbb{R}^{r\times r}$. An optimal solution to

$$\underset{\mathbf{V}:\mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\text{maximize}} \mathbf{Tr}(\mathbf{Z}^\top\mathbf{V})$$

is given by $\hat{\mathbf{V}} = \mathbf{M}\mathbf{N}^\top$, and the objective function has optimal value $\|\mathbf{Z}\|_*$, the nuclear norm of $\mathbf{Z}$.

Proof. See in Appendix 2.9.1. □

To analyze the computational complexity of the algorithm, we see a one-time computation of $\mathbf{Y}^\top\mathbf{X}$ that costs $O(npq)$. In each iteration, there is $O(pqr)$ complexity for the matrix multiplication $\mathbf{Y}^\top\mathbf{X}\mathbf{U}^{(k)}$ and $O(qr^2)$ for computing the SVD and the final solution. Therefore, the per-iteration computational complexity for the V-step is $O(pqr + qr^2)$, or $O(pqr)$ when $p \gg q$.

In the U-step, we are solving a group lasso problem. Computing $\mathbf{Y}\mathbf{V}^{(k+1)}$ takes $O(nqr)$ time. The group-lasso problem can be solved by **glmnet** [Friedman et al., 2010b] with the `mgaussian` family. With coordinate descent, its complexity is $O(\tilde{k}pqn)$, where $\tilde{k}$ is the number of iterations until

convergence and is expected to be small with a reasonable initialization, for example, provided by warm start. Thus, the per-iteration complexity for the U-step is $O(nqr + \tilde{k}npq)$, which is $O(\tilde{k}pqn)$ when $p \gg r$.

Therefore, the overall computational complexity scales at least linearly with the number of features, and will have a large multiplier if the sample size is large as well. While subsampling can effectively reduce the computational cost, in high-dimensional settings, it is critical to have sufficient samples for the quality of estimation. Instead, we seek for computational techniques that can lower the actual number of features involved in expensive iterative computation without giving up any statistical efficiency. Thanks to the induced sparsity by the objective function, we are able to achieve it by variable screening.

2.3.2 Variable Screening for Ultrahigh-Dimensional Problems

In this section, we discuss strategic ways to find a good subset of variables to focus on in the computation that would allow us to reconstruct the full solution easily. In particular, we would like to iterate through the following steps for each λ:

1. **Screen** a strong set S and treat all the left-out variables S^c as null variables that potentially have zero coefficients;
2. **Solve** a significantly smaller problem on the subset of variables S;
3. **Check** an optimality condition to guarantee the constructed full solution $\hat{\mathbf{B}} = (\hat{\mathbf{B}}_S, \hat{\mathbf{B}}_{S^c})$ with $\hat{\mathbf{B}}_{S^c} = 0$ is indeed a valid solution to the original problem. If the condition is not satisfied, go back to the first step with an expanded set S.

Screening Strategies

We have seen Lemma 1 that determines the entry point of any nonzero coefficient on the solution path. Furthermore, there is evidence that the variables entering the model (as one decreases the λ value) tend to have large values by this criterion. Tibshirani et al. [2012] developed on this idea and proposed the strong rules as a sequential variable screening mechanism. The strong rules state that in a standard lasso problem with the model matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p) \in \mathbb{R}^{n \times p}$ and a single response

$\mathbf{y} \in \mathbb{R}^n$, assume $\hat{\beta}(\lambda_{k-1})$ is the lasso solution at λ_{k-1}, then the jth predictor is discarded at λ_k if

$$|\mathbf{x}_j^\top (\mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_{k-1}))| < \lambda_k - (\lambda_{k-1} - \lambda_k). \tag{2.5}$$

The key idea is that the inner product above is almost "non-expansive" in terms of λ. As a result, the KKT condition suggests that the variables to be discarded by (2.5) would have coefficient 0 at λ_k. However it is not a guarantee. The strong rules can fail, though failures occur rarely when $p > n$. In any case, the KKT condition is checked to ensure the exact solution is found. Although Tibshirani et al. [2012] focused mostly on the lasso-type problem, it also suggests extension to general objective functions and penalties. For general objective function $f(\beta)$ with p_j-norm penalty $\|\beta_j\|_{p_j}$ for the jth group, the screening criterion will be based on the dual norm of its gradient $\|\nabla_j f(\beta)\|_{q_j}$ where $1/p_j + 1/q_j = 1$.

Inspired by the general strong rules, we propose three sequential screening strategies for the sparse reduced rank objective (2.2), named after their respective characteristics: Multi-Gaussian, Rank-Less and Fix-V. They are based either on the solution of a relaxed convex problem at the same λ_k or on the exact solution at the previous λ_{k-1}.

- (Multi-Gaussian) Solve the full-rank convex problem at λ_k and use its active set as the candidates for the low-rank settings. The main advantage is that the screening is always stable due to the convexity. However this approach often overselects and brings extra burden to the computation. By assuming a higher rank than necessary, the effective number of responses would become more than that of a low-rank model. As a result, more variables would potentially be needed to serve for an enlarged set of responses.

- (Rank-Less) Find variables that have large $c_j = \|\mathbf{X}_j^T(\mathbf{Y} - \mathbf{X}\mathbf{U}(\lambda_{k-1})\mathbf{V}(\lambda_{k-1})^\top)\|_2$. This is analogous to the strong rules applied to the vanilla multi-response lasso ignoring the rank constraint.

- (Fix-V) Find variables that have large $c'_j = \|\mathbf{X}_j^T(\mathbf{Y}\mathbf{V}(\lambda_{k-1}) - \mathbf{X}\mathbf{U}(\lambda_{k-1}))\|_2$. This is similar to the strong rules applied in the $\mathbf{U}$-step with $\mathbf{V}$ assumed fixed. To see the rationale better, we take another perspective. The squared error in SRRR (2.2) can also be written as

$$\|\mathbf{Y} - \mathbf{X}\mathbf{U}\mathbf{V}^\top\|_F^2 = \operatorname{tr}(\mathbf{Y}^\top\mathbf{Y}) - 2\operatorname{tr}(\mathbf{Y}^\top\mathbf{X}\mathbf{U}\mathbf{V}^\top) + \operatorname{tr}(\mathbf{X}\mathbf{U}\mathbf{V}^\top\mathbf{V}\mathbf{U}^\top\mathbf{X}^\top)$$

Since $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$, the optimization problem becomes

$$\underset{\mathbf{U},\mathbf{V}:\mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\text{minimize}} \frac{1}{2}\|\mathbf{XU}\|_F^2 - \text{tr}(\mathbf{Y}^\top\mathbf{XUV}^\top) + \lambda\sum_{j=1}^{p}\|\mathbf{U}_{j\cdot}\|_2$$

For any given $\mathbf{U}$, we can solve $\mathbf{V} = \mathbf{MN}^\top$, where $\mathbf{Y}^\top\mathbf{XB} = \mathbf{MDN}^\top$ is its singular value decomposition. Let $f(\mathbf{U}) = \frac{1}{2}\|\mathbf{XU}\|_F^2 - \|\mathbf{Y}^\top\mathbf{XU}\|_*$. The problem is reduced to

$$\underset{\mathbf{U}}{\text{minimize}}\, f(\mathbf{U}) + \lambda\sum_{j=1}^{p}\|\mathbf{U}_{j\cdot}\|_2$$

The general strong rule tells us to screen based on the gradient; that is

$$\nabla_{\mathbf{B}} f(\mathbf{B}) = \mathbf{X}^\top\mathbf{XU} - \mathbf{X}^\top\mathbf{YMN}^\top = \mathbf{X}^\top(\mathbf{XU} - \mathbf{YV}).$$

Therefore, the general strong rules endorse the use of this screening rule.

We do some experiments to compare the effectiveness of the rules. We simulate the model matrix under an independent design and an equi-correlated design with correlation $\rho = 0.5$. The true solution path is computed using Algorithm 2 with several random initializations and the convex relaxation-based initialization (as in the Multi-Gaussian rule). Let $S(\lambda)$ be the true active set at λ. For each method ℓ above, we can find, based on either the exact solution at λ_{k-1} or the full-rank solution at λ_k, the threshold it needs so that by the screening criterion, the selected subset $\hat{S}(\lambda_k)^{(\ell)}$ contains the true subset at λ_k, i.e. $\hat{S}(\lambda_k|\lambda_{k-1})^{(\ell)} \supseteq S(\lambda_k)$. This demonstrates how deep each method has to search down the variable list to include all necessary variables, and thus how accurate the screening mechanism is — the larger the subset size, the worse the method is.

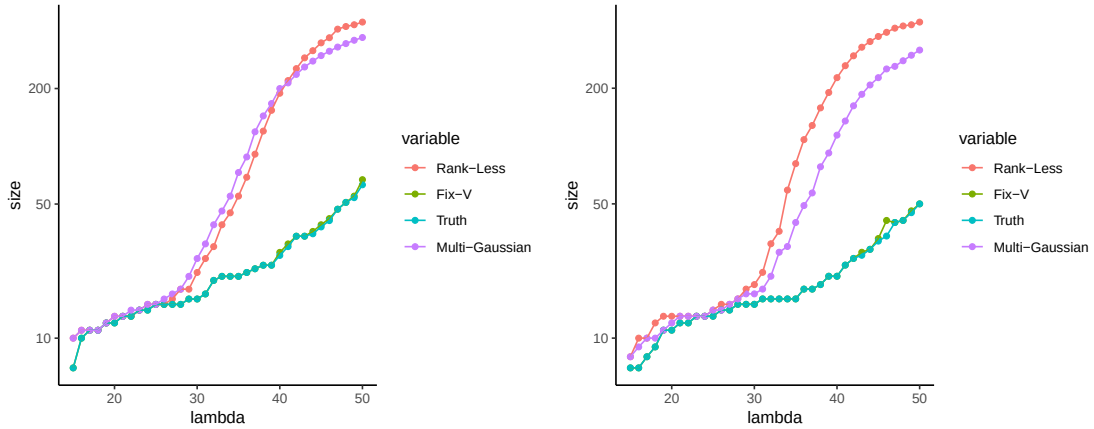


Figure 2.2: Size of screened set under different strategies. Left: independent design. Right: equi-correlated design with $\rho = 0.5$. Signal-to-noise ratio (SNR) $= 1$, and we use the true rank $= 3$.

We see from both plots that the curve of the Fix-V method is able to track that of the exact subset fairly well, while the Rank-Less and Multi-Gaussian methods should choose a much larger subset in order to cover the subset of active variables in the exact solution. In the rest of the paper, we will adopt the Fix-V method to do variable screening.

Optimality Condition

Although the Fix-V method turns out to be most effective in choosing the subset of variables, in practice we have no access to the true subset and have to take an estimate. Instead of trying to find a sophisticated threshold, we will do batch screening at a fixed size (this size can change adaptively though). Given a size K, we will take the K variables that rank the top under this criterion. Clearly we can make mistakes by having left out some important variables in the screening stage. In order to make sure that our solution is exact rather than approximate in terms of the original problem, we need to check the optimality condition and take in more variables when necessary.

Suppose we find a solution $\hat{\mathbf{U}}_S, \hat{\mathbf{V}}_S$ on a subset of variables $\mathbf{X}_S$ by alternating minimization. We will verify the assembled solution $\hat{\mathbf{U}} = (\hat{\mathbf{U}}_S, \mathbf{0}), \hat{\mathbf{V}} = \hat{\mathbf{V}}_S$ is a limiting point of the original optimization problem. The argument is supported by the following lemma.

Lemma 3. In the U-step (2.4), given $\mathbf{V}$ and λ, if we have an exact solution $\hat{\mathbf{U}}_S$ for the sub-problem

with $\mathbf{X}_S$, then $\hat{\mathbf{U}} = (\hat{\mathbf{U}}_S, \mathbf{0})$ is a solution to the full problem if and only if for all $j \in S^c$,

$$\|\mathbf{x}_j^\top (\mathbf{Y}\mathbf{V} - \mathbf{X}_S \hat{\mathbf{U}}_S)\|_2 \leq \lambda. \tag{2.6}$$

Proof. Since this is a convex problem, $\hat{\mathbf{U}}$ is a solution if and only if $\mathbf{0} \in \partial f(\hat{\mathbf{U}})$ where f is the objective function in (2.4) and ∂f is its subdifferential. For the vector ℓ_2-norm, we know that the subdifferential of $\|\mathbf{x}\|_2$ is $\{\mathbf{s} \in \mathbb{R}^p : \|\mathbf{s}\|_2 \leq 1\}$ if $\mathbf{x} = \mathbf{0}$ and $\{\mathbf{x}/\|\mathbf{x}\|_2\}$ if $\mathbf{x} \neq \mathbf{0}$. Notice that $\mathbf{X}_S \hat{\mathbf{U}}_S = \mathbf{X}\hat{\mathbf{U}}$ by the definition of $\hat{\mathbf{U}}$. Since we have an exact solution on S, we know $\mathbf{0} \in \partial f(\hat{\mathbf{U}})_j$. for all $j \in S$. On the other hand, for $j \in S^c$, $0 \in \partial f(\hat{\mathbf{U}})$ if and only if $\mathbf{0} \in \{\mathbf{x}_j^\top (\mathbf{X}\hat{\mathbf{U}} - \mathbf{Y}\mathbf{V}) + \lambda \mathbf{s}_j : \|\mathbf{s}_j\|_2 \leq 1\}$, which is further equivalent to $\|\mathbf{x}_j^\top (\mathbf{Y}\mathbf{V} - \mathbf{X}_S \hat{\mathbf{U}}_S)\|_2 = \|\mathbf{x}_j^\top (\mathbf{Y}\mathbf{V} - \mathbf{X}\hat{\mathbf{U}})\|_2 \leq \lambda$ □

Therefore, once we obtain a solution $\hat{\mathbf{U}}_S, \hat{\mathbf{V}}_S$ for the sub-problem and get condition (2.6) verified, we know in the V-step, by the lemma above, $\hat{\mathbf{U}} = (\hat{\mathbf{U}}_S, \mathbf{0})$ is the solution given $\hat{\mathbf{V}} = \hat{\mathbf{V}}_S$. In the U-step, since $\mathbf{X}\hat{\mathbf{U}} = \mathbf{X}_S \hat{\mathbf{U}}_S$, $\hat{\mathbf{U}}$ is the solution to the full problem. We see that $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$ is a limiting point of the alternating minimization algorithm for the original problem. However if the condition fails, we expand the screened set or bring in the violated variables, and do the fit again. We should note that when we say an exact solution to the original problem, we do not claim it to be a local minimum or global minimum, unless under some regularity conditions as will be briefly discussed later. It is a limiting point of the vanilla alternating minimization algorithm, i.e. Algorithm 2. In other words, if we start from the constructed solution (with zero coefficients for the leftout variables), the algorithm should converge in one iteration and return the same solution.

We have seen the main ingredients of the iterative algorithm: screening, solving and checking. Next we discuss some useful practical considerations and extensions.

2.3.3 Computational Considerations

Initialization and Warm Start

Recall that in the training stage our goal is to fit an SRRR solution path across different λ values. It is easy to see that with a careful choice of parameterization, the path is continuous in λ. To leverage this property, we adopt a warm start strategy. Specifically, we initialize the coefficients of the existing variables at λ_{k+1} using the solution at λ_k and zero-initialize the newly added variables. With warm start, much less iterations will be needed to converge to the new minimum.

However, this by no means guarantees that we are all on a good path. It's likely that we are trapped into a neighborhood of local optimum and end up with much higher function value than the global minimum. One way to alleviate this, if affordable, is to solve the corresponding full-rank problem first, and initialize the coefficients with low-rank approximation of the full-rank solution. We can compare the limiting function values with the warm-start initialization and see which converges to a better point. Although we didn't use in the actual implementation and experiments, one could also do random exploration — randomly initialize some of the coefficients, run the algorithm multiple times and find one that achieves the lowest function value. That said, we lose the advantage of warm start though. The good news is, in the experiments we have done, we didn't observe very clear suboptimal behavior by the warm start and full-rank strategies.

Early Stopping

Although we pre-specify a sequence of λ values $\lambda_1 > \lambda_2 > \cdots > \lambda_L$ where we want to fit the SRRR models, we do not have to fit them all given our goal is to find the best predictive model. Once the model starts to overfit as we move down the λ list, we can stop our process since the later models will have no practical use and are expensive to train. Therefore, in the actual computation, we monitor the validation error along the solution path and call it a stop if it shows a clear upward trend. One other point we would like to make in this regard is that the validation metric can be defined either as an average MSE over all phenotypes or a subset of phenotypes we are most interested in. This is because practically the best λ value can be different for different phenotypes in the joint model.

2.3.4 Extensions

Standardization

We often want to standardize the predictors if they are not on the same scale because the penalty term is not invariant to change of units of the variables. However we emphasize that some thought has to be put into this before standardizing the predictors. If the predictors are already on the same scale, standardizing them could bring unintended advantages to variables with smaller variance themselves. It is more reasonable not to standardize in such cases.

In terms of the outcomes, since they can be at different scales, it is important to standardize

them in the training stage so that no one dominates in the objective function. At prediction (both training and test time), we scale back to the original levels using their respective variances from the training set. In fact, the real impact an outcome has to the overall objective is determined by the proportion of unexplained variance. It would be good to weight the responses properly based on this if such information is available or can be estimated, e.g. via heritability estimation for phenotypes in genetic studies.

Weighting

Sometimes we have strong reasons or evidence to prioritize some of the predictors than the others. We can easily extend the standard objective (2.2) and reflect this belief in a weighted penalty $\lambda \sum_{j=1}^{p} w_j \|\mathbf{U}_{j\cdot}\|_2$ where the weight w_j controls inversely the relative importance of the jth variable. For example, $w_j = 0$ implies jth variable will always be included in the model, while a large w_j will almost exclude the variable from the model.

In the response space, we can also impose a weighting mechanism to priortize the training of certain responses. For a given set of nonnegative weights $w_k, 1 \le k \le q$, the SRRR objective (2.2) can be modified to $(1/2)\sum_{k=1}^{q} w_k \|\mathbf{Y}_{\cdot k} - \mathbf{X}\mathbf{U}\mathbf{V}_{k\cdot}^{\top}\|_2^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2$ with the same constraint, or equivalently,

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{Y}\mathbf{W}^{\frac{1}{2}} - \mathbf{X}\mathbf{U}\mathbf{V}^{\top}\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2, \\ \text{s.t.} \quad & \mathbf{V}^{\top}\mathbf{W}^{-1}\mathbf{V} = \mathbf{I}, \end{aligned} \tag{2.7}$$

where the weight matrix $\mathbf{W} = \text{diag}(w_1, \ldots, w_q)$. To solve the problem with our alternating minimization scheme, we can see that in the V-step, instead of solving the standard orthogonal Procrustes problem with an elegant analytic solution derived from the SVD, we have to deal with a so-called weighted orthogonal Procrustes problem (WOPP). Finding the solution of the WOPP is far more complicated. See, for instance, Mooijaart and Commandeur [1990], Chu and Trendafilov [1998] and Viklands [2006]. An iterative procedure is often needed to compute the solution. For

better computational efficiency, we instead solve the problem with the original orthonormal constraint:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{Y}\mathbf{W}^{\frac{1}{2}} - \mathbf{X}\mathbf{U}\mathbf{V}^\top\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2, \\ \text{s.t.} \quad & \mathbf{V}^\top\mathbf{V} = \mathbf{I}. \end{aligned} \tag{2.8}$$

That is, we amplify the magnitude of some responses so that the objective value is more sensitive to the loss incurred on these responses. When making prediction, we will need to scale them back to the original units.

Adjustment Covariates

In some applications such as genome-wide association studies (GWAS), there may be confounding variables $\mathbf{Z} \in \mathbb{R}^{n\times m}$ that we want to adjust for in the model. For example, population stratification, defined as the existence of a systematic ancestry difference in the sample data, is one of the common factors in GWAS that can lead to spurious discoveries. This can be controlled for by including some leading principal components of the SNP matrix as variables in the regression [Price et al., 2006]. In the presence of such variables, we solve the following problem instead. With a slight abuse of notation, in this section, we use $\mathbf{W}$ to denote the coefficient matrix for the covariates instead of a weight matrix:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{Y} - \mathbf{Z}\mathbf{W} - \mathbf{X}\mathbf{U}\mathbf{V}^\top\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}^j\|_2, \\ \text{s.t.} \quad & \mathbf{V}^\top\mathbf{V} = \mathbf{I}. \end{aligned} \tag{2.9}$$

The main components don't change except two adjustments. When determining the starting λ value, we use Lemma 4.

Lemma 4. In problem (2.9), if $r > 0$, the maximum λ that results in a nontrivial solution $\hat{B}(\lambda)$ is

$$\lambda_{\max} = \max_{1\le j\le p} \|\mathbf{x}_j^\top \hat{\mathbf{R}}\|_2,$$

where $\hat{\mathbf{R}} = \mathbf{Y} - \mathbf{Z}\hat{\mathbf{W}}$ and $\hat{\mathbf{W}}$ is the multiple outcome regression coefficient matrix.

The proof is almost the same as before. The other nuance we should be careful about is when fitting the model, we should leave those covariates unpenalized because they serve for the adjustment purpose and should not be experiencing the selection stage. In particular, in the U-step (group lasso) given $\mathbf{V}$, direct computation would reduce to solving the problem

$$\underset{\mathbf{U},\mathbf{W}}{\text{minimize}} \qquad \frac{1}{2}\|\mathbf{YV} - \mathbf{ZWV} - \mathbf{XU}\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}^j\|_2,$$

which is not as convenient as standard group lasso problem. Instead, we find that $\mathbf{W}$ can always be solved explicitly in terms of other variables. In fact, the minimizer $\hat{\mathbf{W}} = (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top(\mathbf{Y} - \mathbf{XUV}^\top)$. Plug in and we find that the problem to be solved can be written as

$$\underset{\mathbf{U}}{\text{minimize}} \qquad \frac{1}{2}\|(\mathbf{I} - \mathbf{H_Z})\mathbf{YV} - (\mathbf{I} - \mathbf{H_Z})\mathbf{XU}\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}^j\|_2,$$

where $\mathbf{H_Z} = \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top$ is the projection matrix on the column space of $\mathbf{Z}$. This becomes a standard group lasso problem and can be solved by using, for example, the **glmnet** package with the `mgaussian` family.

Missing Values

In practice, there can be missing values in either the predictor matrix or the outcome matrix. If we only discard samples that have any missing value, we could lose a lot of information. For the predictor matrix, we could do imputation as simple as mean imputation or something sophisticated by leveraging the correlation structure. For missingness in the outcome, there is a natural way to integrate an imputation step seamlessly with the current procedure, analogous to the softImpute idea in Mazumder et al. [2010]. We first define a projection operator for a subset of two dimensional indices $\Omega \subseteq \{1, \ldots, n\} \times \{1, \ldots, p\}$. Let $\mathcal{P}_\Omega : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}$ be such that

$$\mathcal{P}_\Omega(\mathbf{Y})_{i,j} = \begin{cases} \mathbf{Y}_{i,j}, & (i,j) \in \Omega, \\ 0, & (i,j) \notin \Omega. \end{cases}$$

Let Ω be the set of indices where the response values are observed; in other words, Ω^c is the set of missing locations. Instead of (2.2), now we solve the following problem.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{X}\mathbf{U}\mathbf{V}^\top)\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}^j\|_2, \\ \text{s.t.} \quad & \mathbf{V}^\top \mathbf{V} = \mathbf{I}. \end{aligned} \tag{2.10}$$

We can easily see that an equivalent formulation of the problem is

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V},\mathbf{Y}'}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{Y}' - \mathbf{X}\mathbf{U}\mathbf{V}^\top\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2, \\ \text{s.t.} \quad & \mathbf{V}^\top \mathbf{V} = \mathbf{I},\ \mathcal{P}_\Omega(\mathbf{Y}') = \mathcal{P}_\Omega(\mathbf{Y}). \end{aligned}$$

This inspires a natural projection step to deal with the additional constraint. It can be well integrated with the current alternating minimization scheme. In fact, after each alternation between the U-step and the V-step, we can impute the missing values from the current predictions $\mathbf{X}\mathbf{U}\mathbf{V}^\top$, and then continue into the next U-V alternation with the completed matrix.

Lazy Reduced Rank Regression

There is an alternative way to find a low-rank coefficient profile for the multivariate regression. Instead of pursing to solve the non-convex problem (2.2) directly, we can follow a two-stage procedure:

1. Solve a full-rank multi-gaussian sparse regression, i.e.,

$$\text{minimize}_{\mathbf{B}} \quad \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\mathbf{B}\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{B}_{j\cdot}\|_2.$$

2. Conduct SVD of the resulting coefficient matrix $\hat{\mathbf{B}}$ and use its rank r approximation as our final estimator.

The advantage of this approach is that it is stable. The first stage is a convex problem and can be handled efficiently by, for example, `glmnet`. A variety of adaptive screening rules are also applicable in this situation to assist dimension reduction. The second stage is fairly standard and efficient as long as there are not too many active variables. However, the disadvantage is clear too. The

low-rank approximation is conducted in an unsupervised manner, so could lead to some degrade in the prediction performance.

That said, as before, we should still evaluate the out-of-sample performance as the penalty parameter λ varies and pick the best on the solution path as our final estimated model. In many cases, we compute the full-rank model under the exact mode anyways, so the set of lazy models can be thought of as an efficient byproduct for our choice.

2.3.5 Full Algorithm

We incorporate the options above and present the full algorithm in Algorithm 3.

Algorithm 3 Large-scale and Ultrahigh-dimensional Sparse Reduced Rank Regression

Standardize or weight the responses. Define a sequence of λ values $\lambda_1 > \cdots > \lambda_L$. Initialize $\mathbf{U}(\lambda_0) = \mathbf{0}, \mathbf{V}(\lambda_0) = \mathbf{0}$ and $\mathbf{Y}_{\Omega^c}$.

for $\ell = 1$ **to** L **do**

  Initialize $t = 0$, $\mathbf{U}(\lambda_\ell) = \mathbf{U}(\lambda_{\ell-1}), \mathbf{V}(\lambda_\ell) = \mathbf{V}(\lambda_{\ell-1}), \mathbf{W}(\lambda_\ell) = \mathbf{W}(\lambda_{\ell-1})$, and $\mathcal{A}(\lambda_\ell)$ be the active set at $\lambda_{\ell-1}$.

  while $t = 0$ **or** KKT Check at $t-1$ failed **do**

    [Variable Screening] Find M variables $S_M \subseteq \Omega \setminus \mathcal{A}(\lambda_\ell)$ with largest values in $\|\mathbf{x}_j^\top(\mathbf{Y} - \mathbf{ZW}(\lambda_\ell) - \mathbf{X}_{\mathcal{A}(\lambda_\ell)}\mathbf{U}_{\mathcal{A}(\lambda_\ell)}(\lambda_\ell)\mathbf{V}(\lambda_\ell)^\top\|$, and let

$$\mathcal{A}(\lambda_\ell) = \mathcal{A}(\lambda_\ell) \cup S_M.$$

    [Alternating Minimization] Let $k = 0$ and $\mathbf{U}^{(0)} = \mathbf{U}_{\mathcal{A}(\lambda_\ell)}(\lambda_\ell), \mathbf{V}^{(0)} = \mathbf{V}(\lambda_\ell), \mathbf{W}^{(0)} = \mathbf{W}(\lambda_\ell)$ and $\mathbf{Y}^{(0)} = \mathbf{Y}$.

    while $k = 0$ **or** $\|\mathbf{U}^{(k)}{\mathbf{V}^{(k)}}^\top - \mathbf{U}^{(k-1)}{\mathbf{V}^{(k-1)}}^\top\| > \epsilon$ **do**

      V-step: Fix $\mathbf{U}^{(k)}$, solve $\mathbf{V}$: the orthogonal Procrustes problem

$$\underset{\mathbf{V}:\mathbf{V}^\top\mathbf{V}=\mathbf{I}}{\text{minimize}} \ \|\mathbf{Y}^{(k)} - \mathbf{ZW}^{(k)} - \mathbf{X}_{\mathcal{A}(\lambda_\ell)}\mathbf{U}^{(k)}\mathbf{V}^\top\|_F^2.$$

      Let $(\mathbf{Y}^{(k)} - \mathbf{ZW}^{(k)})^\top\mathbf{X}_{\mathcal{A}(\lambda_\ell)}\mathbf{U}^{(k)} = \mathbf{MDN}^\top$ (skinny SVD) and solve $\mathbf{V}^{(k+1)} = \mathbf{MN}^\top$.

      U-step: Fix $\mathbf{V}^{(k+1)}$, solve $\mathbf{U}$ and $\mathbf{W}$: the group lasso problem

$$\mathbf{U}^{(k+1)} = \text{argmin}_{\mathbf{U}} \ \frac{1}{2}\|(\mathbf{I}-\mathbf{H}_\mathbf{Z})\mathbf{Y}^{(k)}\mathbf{V}^{(k+1)} - (\mathbf{I}-\mathbf{H}_\mathbf{Z})\mathbf{X}_{\mathcal{A}(\lambda_\ell)}\mathbf{U}\|_F^2 + \lambda_\ell \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2, \quad (2.11)$$

      and $\mathbf{W}^{(k+1)} = (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top(\mathbf{Y}^{(k)} - \mathbf{X}_{\mathcal{A}(\lambda_\ell)}\mathbf{U}^{(k+1)}\mathbf{V}^{(k+1)})$.

      Y-step: Impute the missing values

$$\mathbf{Y}_\Omega^{(k+1)} = \mathbf{Y}_\Omega^{(k)}, \ \mathbf{Y}_{\Omega^c}^{(k+1)} = (\mathbf{ZW}^{(k+1)} + \mathbf{X}_{\mathcal{A}(\lambda_\ell)}\mathbf{U}^{(k+1)}(\mathbf{V}^{(k+1)})^\top)_{\Omega^c}$$

      $k = k + 1$

    end while

    Let $\mathbf{U}_{\mathcal{A}(\lambda_\ell)}(\lambda_\ell) = \mathbf{U}^{(k)}, \mathbf{U}_{\mathcal{A}(\lambda_\ell)}(\lambda_\ell) = \mathbf{0}, \mathbf{V}(\lambda_\ell) = \mathbf{V}^{(k)}, \mathbf{W}(\lambda_\ell) = \mathbf{W}^{(k)}$ and $\mathbf{Y} = \mathbf{Y}^{(k)}$.

    [KKT Check] Check the criterion for all $j \in \Omega \setminus \mathcal{A}(\lambda_\ell)$,

$$\|\mathbf{x}_j^\top(\mathbf{Y} - \mathbf{ZW}(\lambda_\ell) - \mathbf{X}_{\mathcal{A}(\lambda_\ell)}\mathbf{U}_{\mathcal{A}(\lambda_\ell)}(\lambda_\ell)\mathbf{V}(\lambda_\ell)^\top)\| \le \lambda_\ell.$$

    $t = t + 1$

  end while

2.4 Convergence Analysis

In this section, we present some convergence properties of the alternating minimization algorithm (Algorithm 2) on sparse reduced rank regression. Let

$$g(\mathbf{U}, \mathbf{V}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\mathbf{U}\mathbf{V}^\top\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathbf{U}^j\|_2.$$

Theorem 1. For any $k \geq 1$, the function values are monotonically decreasing:

$$g(\mathbf{U}^k, \mathbf{V}^k) \geq g(\mathbf{U}^{k+1}, \mathbf{V}^k) \geq g(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}).$$

Furthermore, we have the following finite convergence rate:

$$\min_{1 \leq k \leq K} g(\mathbf{U}^k, \mathbf{V}^k) - g(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}) \leq \frac{1}{K}(g(\mathbf{U}^1, \mathbf{V}^1) - g^\infty),$$

where $g^\infty = \lim_{k \to \infty} g(\mathbf{U}^k, \mathbf{V}^k)$. It implies that the iteration will terminate in $O(1/\epsilon)$ iterations.

The proof is straightforward and we won't detail here. It presents the fact that alternating minimization is a descent algorithm. In fact, this property holds for all alternating minimization or more general blockwise coordinate descent algorithms. However it does not say how good the limiting point is. In the next result, we show a local convergence result that under some regularity conditions, if the initialization is closer enough to a global minimum, it will converge to a global minimum at linear rate. It is based on similar results on proximal gradient descent by Dubois et al. [2019]. To define a local neighborhood, it would be easier if we eliminate $\mathbf{V}$ by always setting it to a minimizer given $\mathbf{U}$. That is, the objective function becomes $F_\lambda(\mathbf{U}) = \frac{1}{2}\|\mathbf{X}\mathbf{U}\|_2^2 - \|\mathbf{Y}^\top\mathbf{X}\mathbf{U}\|_* + \lambda \sum_{j=1}^{p} \|\mathbf{U}_{j\cdot}\|_2$. We define a sublevel set $\mathcal{S}_c(\lambda) = \{\mathbf{U} \in \mathbb{R}^{p \times r} : F_\lambda(\mathbf{U}) \leq c\}$.

Theorem 2. Assume $\mathbf{X}^\top\mathbf{X}$ is invertible and $\sigma_{\max}^2 \geq \sigma_{\min}^2 > 0$ be its smallest and largest eigenvalues. Let s_j be the jth singular value of $(\mathbf{X}^\top\mathbf{X})^{-\frac{1}{2}}\mathbf{X}^\top\mathbf{Y}$. There exists $\bar{\lambda} > 0$ such that for all $0 \leq \lambda < \bar{\lambda}$ and $0 \leq \mu < \sigma_{\min}^2(1 - s_{r+1}^2/s_r^2)$, there is a sublevel set $\mathcal{S}(\lambda, \mu)$ where the level depends on λ and μ such that if $\mathbf{U}^k \in \mathcal{S}(\lambda, \mu)$, we have

$$\Delta(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}) \leq \left(1 - \min\left(\frac{1}{2}, \frac{\mu}{\sigma_{\max}^2}\right)\right) \Delta(\mathbf{U}^k, \mathbf{V}^k),$$

where $\Delta(\mathbf{U}, \mathbf{V}) = g(\mathbf{U}, \mathbf{V}) - g(\mathbf{U}^*, \mathbf{V}^*)$ and $(\mathbf{U}^*, \mathbf{V}^*)$ is a global minimum.

From a high level, the proof is based on the fact that under the conditions, the function is strongly convex near the global minima. If we starting from this region, we achieve good convergence rate with alternating minimization algorithm. The full proof is given in Appendix 2.9.1.

It is easy to see that the theorem above implicitly assumes the classical setting where $n \geq p$ since otherwise $\mathbf{X}^\top \mathbf{X}$ would not be invertible. However, it is still applicable to our algorithm. The algorithm does not attempt to solve alternating minimization at the full scale, but only does it after variable screening. With screening, it is very likely that we will again be working under the classical setting. Moreover, with warm start, there is higher chance that the initialization lies in the local region as defined above. Therefore, this theorem can provide useful guidance on the practical computational performance of the algorithm.

2.5 Simulation Studies

We conduct some experiments to gain more insight into the method and compare with the single-response lasso method. Due to space limit, we demonstrate the results in one experiment setting and include results for other settings such as correlated features, deviation from the true low-rank structure etc., in Appendix 2.9.3. We experiment with three different sizes and three different signal-to-noise ratio (SNR): $(n, p, k) = (200, 100, 20), (200, 500, 20), (200, 500, 50)$, where k is the number of variables with true nonzero coefficients, and the target SNR $= 0.5$, 1, or 3. The number of responses $q = 20$ and the true rank $r = 3$. We generate the $\mathbf{X} \in \mathbb{R}^{n \times p}$ with independent samples from some multivariate Gaussian $\mathcal{N}(0, \Sigma_X)$ where $\Sigma_X = \mathbf{I}_p$ in this section. More results under correlated designs are presented in the appendix. The response is generated from the true model $\mathbf{Y} = \mathbf{X}\mathbf{U}\mathbf{V}^\top + \mathbf{E}$, where each entry in the support of $\mathbf{U} \in \mathbb{R}^{p \times r}$ (sparsity k) is independently drawn from a standard Gaussian distribution, and $\mathbf{V} \in \mathbb{R}^{q \times r}$ takes the left singular matrix of a Gaussian ensemble. Hence $\mathbf{B} = \mathbf{U}\mathbf{V}^\top$ is the true coefficient matrix. The noise matrix is generated from $\mathcal{N}(0, \sigma_e^2 \mathbf{I}_q)$, where σ_e^2 is chosen such that the signal-to-noise ratio

$$\text{SNR} = \frac{\text{tr}(\mathbf{B}^\top \Sigma_X \mathbf{B})}{\sigma_e^2 \cdot \text{tr}(\Sigma_E)} \tag{2.12}$$

is set to a given level. The performance is evaluated by the test R^2, defined as follows:

$$R^2 = 1 - \frac{\|\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}\|_F^2}{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_F^2}.$$

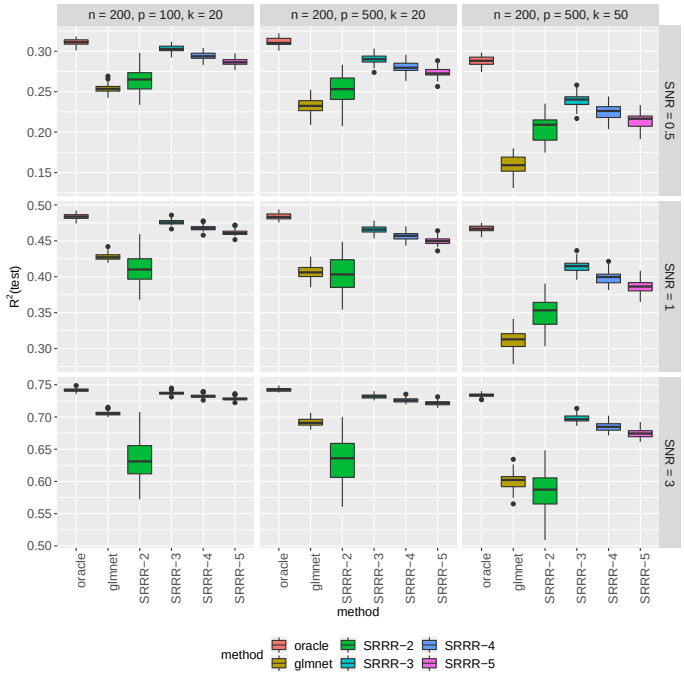


Figure 2.3: R^2 each run is evaluated on a test set of size 5000. "oracle" is the result where we know the true active variables and solve on this subset of variables. "glmnet" fits the responses separately. "SRRR-r" indicates the SRRR results with assumed rank r.

The main insight we obtain from the experiments is that the method is more robust to overestimating than underestimating the rank. A significant degrade in performance can be identified even if we are only off the rank by 1 from below. In contrast, the additional variance brought along by overestimating the rank doesn't seem to be a big concern. This, in essence, can be ascribed to bias and variance decomposition. In our settings, the bias incurred in underestimating the rank and thus 1/3 loss of parameters contributes a lot more to the MSE compared with the increased variance due to 1/3 redundancy in the parameters.

2.6 Real Data Application: UK Biobank

The UK Biobank [Bycroft et al., 2018] is a large, prospective population-based cohort study with individuals collected from multiple sites across the United Kingdom. It contains extensive genetic and phenotypic detail such as genome-wide genotyping, questionnaires and physical measures for a wide range of health-related outcomes for over 500,000 participants, who were aged 40-69 years when recruited in 2006-2010. In this study, we are interested in the relationship between an individual's genotype and his/her phenotypic outcomes. While genome-wide association studies (GWAS) focus on identifying SNPs that may be marginally associated with the outcome using univariate tests, we would like to leverage the additive effect of all SNPs to make good prediction. Recently there is a line of work [Qian et al., 2019, Sinnott-Armstrong et al., 2019, Lello et al., 2018] that builds a lasso solution on the large dataset and shows that the prediction is much improved over previous methods. Furthermore, a number of phenotypes present nontrivial correlation structures and we would like to further improve the prediction and stabilize the variable selection by building a joint model for multiple outcomes.

We focused on 337,199 White British unrelated individuals out of the full set of over 500,000 from the UK Biobank dataset [Bycroft et al., 2018] that satisfy the same set of population stratification criteria as in DeBoever et al. [2018]. Each individual has up to 805,426 measured variants, and each variant is encoded by one of the four levels where 0 corresponds to homozygous major alleles, 1 to heterozygous alleles, 2 to homozygous minor alleles and NA to a missing genotype. In addition, we have available covariates such as age, sex, and forty pre-computed principal components of the SNP matrix. Among them, we use age, sex and the top 10 PCs for the adjustment of population stratification [Price et al., 2006].

There are binary responses in the data such as many disease outcomes. Although in principle we can solve for a mixture of Gaussian and binomial likelihood using Newton's method, for ease of computation in this large-scale setting, it is a reasonable approximation to treat them as continuous responses and fit the standard SRRR model. However, after the model is fit, we will refit a logistic regression on the predicted score to obtain a probability estimation. Notice that the refit is still trained on the training set at each λ value.

The number of samples is large in the UK Biobank dataset, so we afford to set aside an independent validation set without resorting to costly cross-validation to find an optimal regularization

parameter. We also leave out a subset of observations as test set to evaluate the final model. In particular, we randomly partition the original dataset so that 70% is used for training, 10% for validation and 20% for test. The solution path is fit on the training set, whereas the desired regularization is selected on the validation set, and the final model is evaluated on the test set.

In the experiment, we compare the performance of the multivariate-response SRRR model with the single-response lasso model. To fit the lasso model, we rely on fast implementation of the **snpnet** package [Qian et al., 2019], and we also refer to the lasso results as snpnet in the results section. For continuous responses, we evaluate the prediction by R-squared (R^2). Given a linear coefficient vector $\hat{\beta}$ (fitted on the training set) and a subset of data $\{(x_i, y_i), 1 \le i \le n\}$, it is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - x_i^\top \hat{\beta})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}.$$

We compute R^2 respectively on the training, validation and test sets. For binary response, misclassification error could be used but it would depend on the calibration. Instead the receiver operating characteristic (ROC) curve provides more information and demonstrates the tradeoff between true positive and false positive rates under different thresholds. The area under the curve (AUC) computes the area under the ROC curve — a larger value indicates a generally better classifier. Therefore, we will evaluate AUCs on the training, validation and test sets for binary responses. When comparing different methods, we evaluate both absolute change and relative change over the baseline method (in particular the already competitive lasso in our case), where the relative change for a given metric is defined as $(\text{metric}_{\text{new}} - \text{metric}_{\text{lasso}})/|\text{metric}_{\text{lasso}}|$.

Computationally, in the UK Biobank experiments, the SNP data are stored in a compressed PLINK format with two-bit encodings. PLINK 2.0 [Chang et al., 2015] provides an extensive set of efficient operations including very fast, multithreaded matrix multiplication. In particular, this matrix multiplication module is heavily used in the steps of screening and KKT check in this work and other lasso-based results [Li et al., 2020, Qian et al., 2019] on the UK Biobank.

2.6.1 Asthma and 7 Blood Biomarkers

Here, we defined asthma based on a mixture of self-reported questionnaire data and hospital inpatient record data described in DeBoever et al. [2018], Tanigawa et al. [2019]. Furthermore, we focused on 7 additional blood count measurements from Category 100081 in UK Biobank containing

results of haematological assays that were performed on whole blood.

We apply the SRRR to the set of phenotypes and expect some performance improvement by leveraging the correlation structure. Choice of the phenotypes: monocyte count, neutrophill count, eosinophill count, basophill count, forced vital capacity (FVC), peak expiratory flow (PEF), and forced expiratory volume in 1 second (FEV1).

Overall, we see small rank representation can maintain predictive power for specific phenotypes (see Figure 2.4) and that overall the multiresponse model improves the prediction over the single-response lasso model (see Figure 2.5).

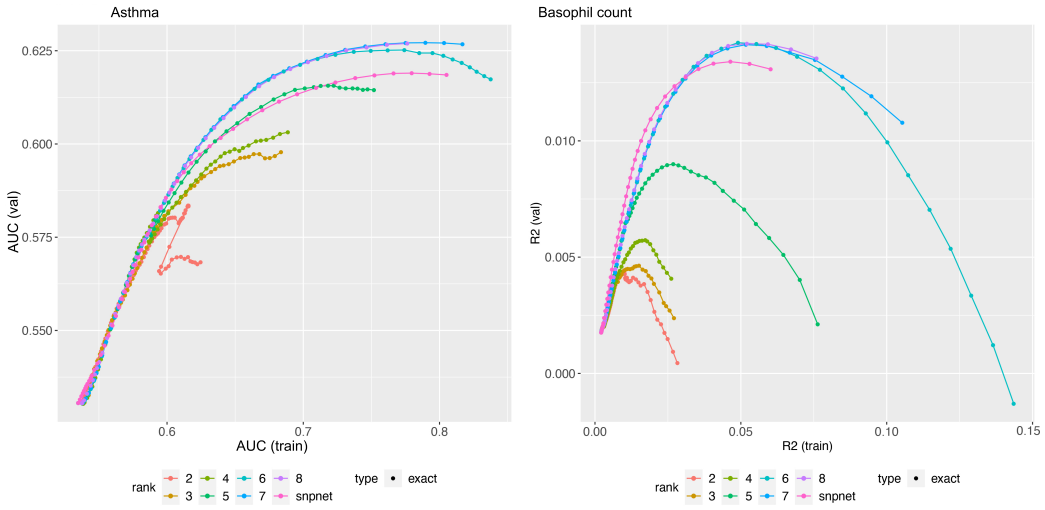


Figure 2.4: Asthma and Basophil count prediction performance plots. Different colors correspond to lower rank predictive performance across (x-axis) training data set and (y-axis) validation data set for (left) asthma and (right) basophil count.

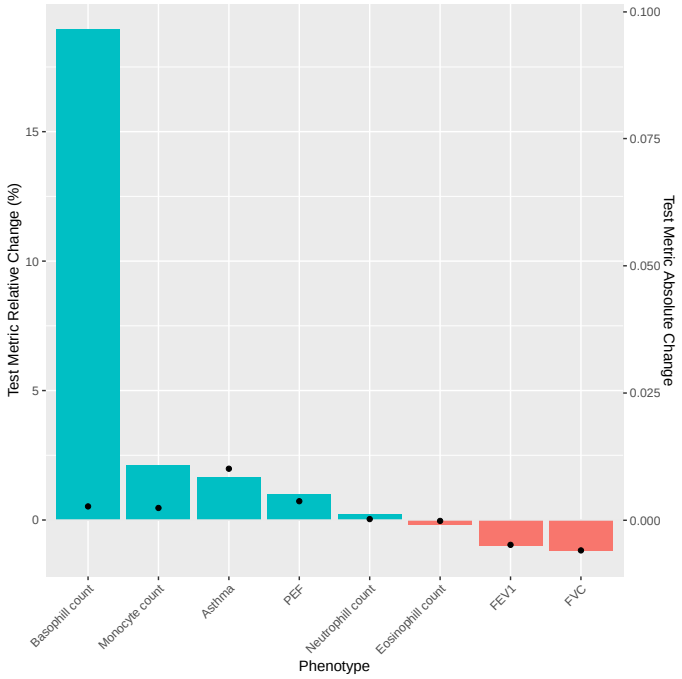


Figure 2.5: Change in prediction accuracy for multiresponse model compared to single response model. (top) (y-axis 1 bar) R^2 relative change (%) for each phenotype (x-axis) and R^2 absolute change (y-axis 2).

2.6.2 35 Biomarkers

In addition, we used 35 biomarkers from the UK Biobank biomarker panel in Sinnott-Armstrong et al. [2019], and apply SRRR to the dataset. Noticeably, for the liver biomarkers including alanine aminotransferase and albumin, and the urinary biomarkers including Microalbumin in urine and Sodium in urine, we see an improvement in prediction performance for the SRRR application beyond the single-response snpnet models (see Figures 2.6 and 2.7).

We can represent the lower rank representation as a biplot of the singular value decomposition of the coefficient matrix [Gower et al., 2011, Gabriel, 1971, Tanigawa et al., 2019]. Specifically, we display phenotypes projected on phenotype principal components as a scatter plot. We also show variants projected on variant principal components as a separate scatter plot and added phenotype singular vectors as arrows on the plot using sub-axes. In scatter plot with biplot annotation, the inner product of a genetic variant and a phenotype represents the direction and the strength of the

projection of the genetic association of the variant-phenotype pair on the displayed latent components. For example, when a variant and a phenotype share the same direction on the annotated scatter plot, that means the projection of the genetic associations of the variant-phenotype pair on the displayed latent components is positive. When a variant-phenotype pair is projected on the same line, but on the opposite direction, the projection of the genetic associations on the shown latent components is negative. When the variant and phenotype vectors are orthogonal or one of the vectors are of zero length, the projection of the genetic associations of the variant-phenotype pair on the displayed latent components is zero. We focused on the top five key SRRR components for AST to ALT ratio (see Figure 2.8).

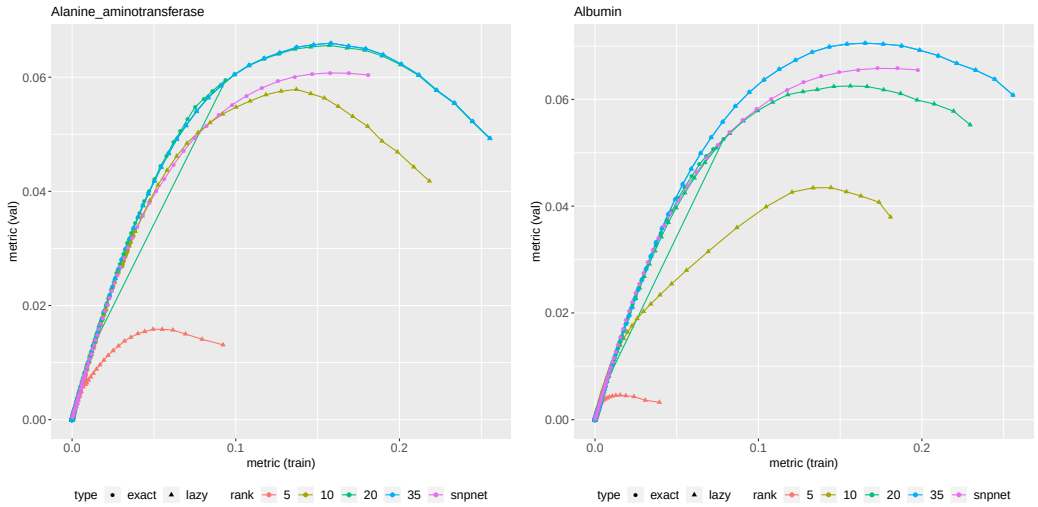


Figure 2.6: Alanine aminotransferase and albumin prediction performance plots. Different colors correspond to lower rank predictive performance across (x-axis) training data set and (y-axis) validation data set for (left) alanine aminotransferase and (right) albumin. For lower rank representation we applied lazy rank evaluation.

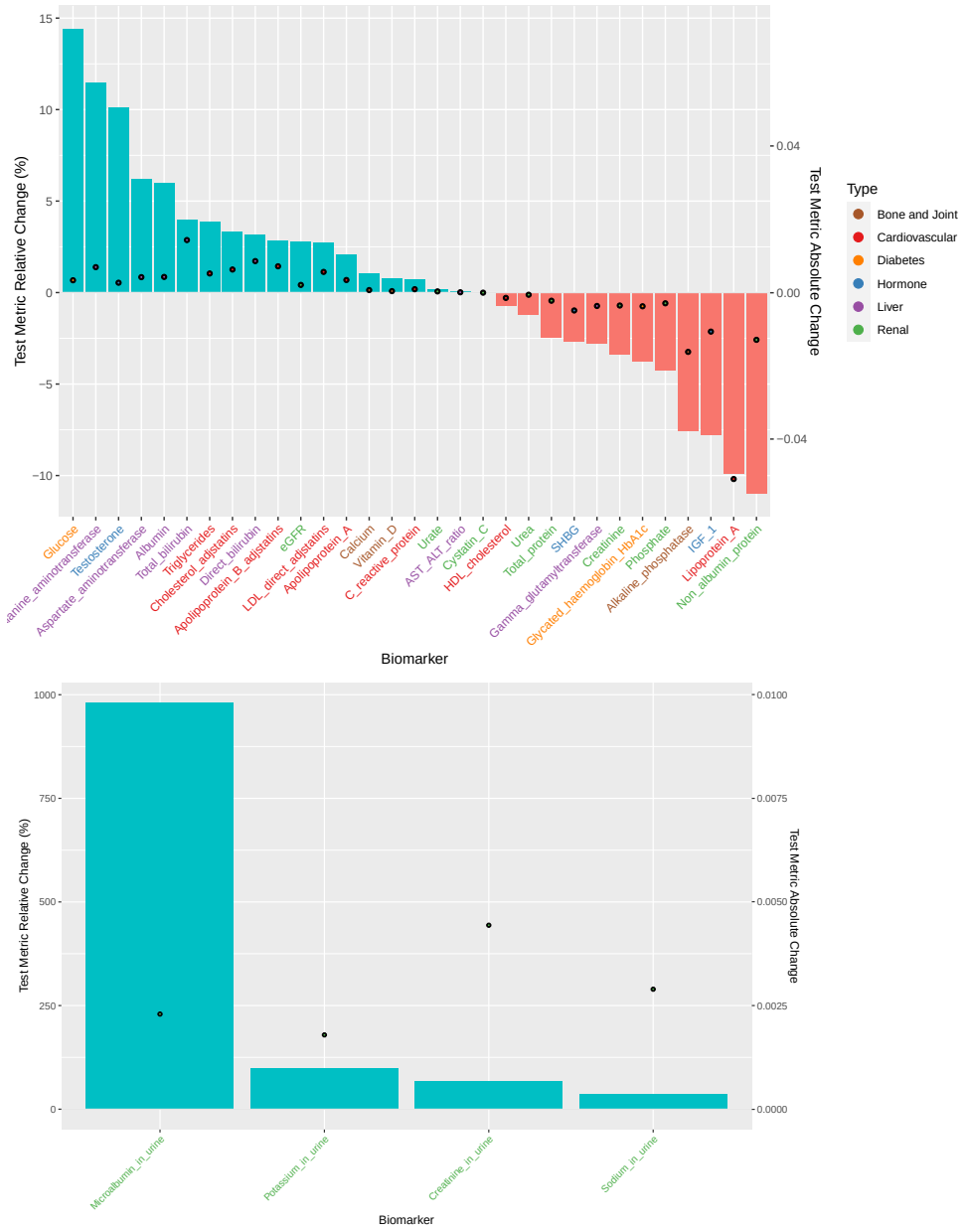


Figure 2.7: Change in prediction accuracy for multiresponse model compared to single response model. (top) (y-axis 1 bar) R^2 relative change (%) for each biomarker (x-axis) across different biomarker category (color) and R^2 absolute change (y-axis 2). (bottom) Change in predictive accuracy for multiresponse model compared to single response model for urinary biomarkers.

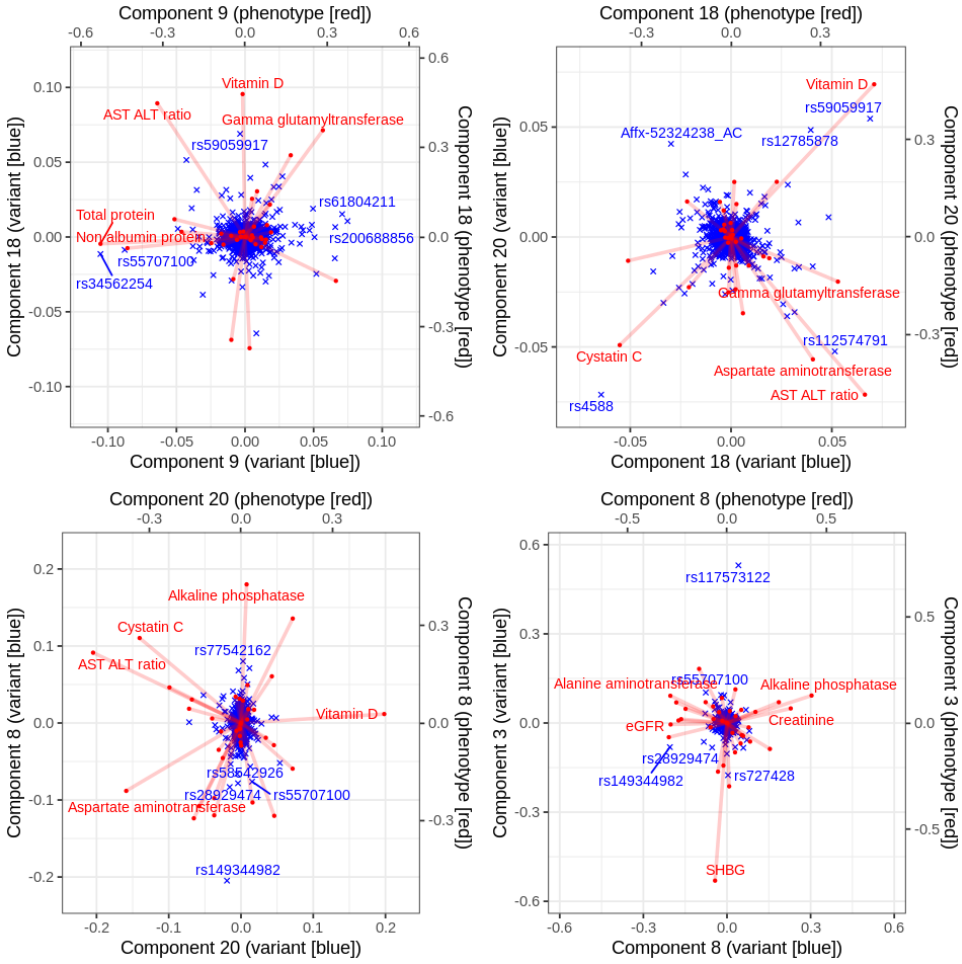


Figure 2.8: The latent structures of the the top five key SRRR components for AST to ALT ratio. Using trait squared cosine score described in Tanigawa et al. [2019], the top five key SRRR components for AST to ALT ratio (components 9, 18, 20, 8, and 3) are identified from a full-rank SVD of coefficient matrix C from SRRR ($C = UDV^T$) and shown as a series of biplots. In each panel, principal components of genetic variants (rows of UD) are shown in blue as scatter plot using the main axis and singular vectors of traits (rows of V) are shown in red dots with lines using the secondary axis, for the identified key components. The five traits and variants with the largest distance from the center of origin are annotated with their name.

2.7 Related Work

There are many other methods that were proposed for multivariate regression in high-dimensional settings. Chen and Huang [2012] compares the SRRR with rank-free methods including L_2SVS

Similä and Tikka [2007], L_∞SVS [Turlach et al., 2005] that replaces the ℓ_2-norm with ℓ_∞-norm of each row, and RemMap [Peng et al., 2010] that imposes an additional elementwise sparsity of the coefficient matrix. It also compares with the SPLS Chun and Keleş [2010] and points out that the latter does not target directly on prediction of the responses so the performance turns out not as good. Another important category of methods Canonical Correlation Analysis (CCA) [Hotelling, 1936] that tries to constructed uncorrelated components in both the feature space and the response space to maximize their correlation coefficients also falls short in the aspect, even though some connection can be established with the reduced rank regression as seen in Appendix 2.9.2.

More recently, there is a line of new advances in sparse and low-rank regression problems. For example, Ma and Sun [2014] proposed a subspace assisted regression with row sparsity and studied its near-optimal estimation properties. Ma et al. [2020] furthered this work to a two-way sparsity setting, where nonzero entries are present only on a few rows and columns. Li et al. [2019] proposed an integrative multi-view reduced-rank regression that encourages group-wise low-rank coefficient matrices with a composite nuclear norm. Dubois et al. [2019] developed a fast first-order proximal gradient algorithm on the SRRR objective reparameterized by a single matrix and proves linear local convergence. Luo et al. [2018] proposed a mixed-outcome reduced-rank regression method that deals with different types of responses and also missing data, though it does not aim for high-dimensional settings with variable selection.

In genetics, some approaches proposed to decompose genetic associations from summary level data using LD-pruning along with p-value thresholding for variable selection in an approach referred to as DeGAs [Tanigawa et al., 2019] and MetaPhat [Lin et al., 2019]. DeGAs was extended for genetic risk prediction and to "paint" an individual's risk to a disease based on genetic component loadings in an approach referred to as DeGAs-risk [Aguirre et al., 2019].

2.8 Summary and Discussion

In this paper, we propose a method that takes into account both sparsity in high-dimensional regression problems and low-rank structure when multiple correlated outcomes are present. A screening-based alternating minimization algorithm is designed to deal with large-scale and ultrahigh-dimensional applications, such as the UK Biobank population cohort. We demonstrate the effectiveness of the

method on both synthetic and real datasets focusing on asthma and 7 related blood count biomarkers, in addition to the 35 biomarker panel made available by UK Biobank [Sinnott-Armstrong et al., 2019]. We anticipate that the approach presented here will generalize to thousands of phenotypes that are currently being measure in UK Biobank, e.g. metabolomics and imaging data that are currently being generated in over 100,000 individuals.

Methodologically, in the UK Biobank experiments, we use continuous approximation to binary outcomes. This is a reasonable assumption but ideally one would like to solve the exact problem based on their respective likelihood. In principle, there is no theoretical challenge in the algorithmic design. We can use Newton's method and enclose the procedure with an outer loop that conducts quadratic approximation of the objective function. However, the quadratic problem involving both penalty and low-rank constraint can be very messy. We might need some heuristics to find a more convenient approximation. We see this as future work along with extending the SRRR algorithm to other families including time-to-event multiple responses that can be used for survival analysis. Furthermore, for an individual we can project a variant and phenotype loading across the reduced rank to their risk to arrive at a similar analysis of outlier individuals with unusual painting of genetic risk and to quantify the overall contribution of a component which may aid in disease risk interpretation. Overall, we see the method and algorithms presented here as an important toolkit to the prediction problem in human genetics.

2.9 Appendix

2.9.1 Additional Proofs

Proof of Lemma 1

This is intuitively the same as one without the rank constraint because when the coefficients just start to become nonzero, the coefficient matrix is low-rank in its nature. Therefore, for the purpose of finding the maximum meaningful λ, we can ignore the rank constraint unless $r = 0$. Without the constraint, it follows from the KKT condition that having all coefficients to be zero is equivalent to setting

$$\lambda \geq \lambda_{\max} = \max_{1 \leq j \leq p} \|\mathbf{x}_j^\top \mathbf{Y}\|_2. \tag{2.13}$$

Therefore, the maximum λ that accommodates a nontrivial solution is $\lambda_{\max} = \max_{1 \leq j \leq p} \|\mathbf{x}_j^\top \mathbf{Y}\|_2$.

Proof of Lemma 2

We plug in the SVD of $\mathbf{Z}$ and have $\mathbf{Tr}(\mathbf{Z}^\top \mathbf{V}) = \mathbf{Tr}(\mathbf{NDM}^\top \mathbf{V}) = \mathbf{Tr}(\mathbf{DM}^\top \mathbf{VN}) = \sum_{k=1}^r \mathbf{D}_{kk}\mathbf{S}_{kk}$, where $\mathbf{S} = \mathbf{M}^\top \mathbf{VN}$ and the last equality is due to the fact that $\mathbf{D}$ is a diagonal matrix. Notice that by the skinny SVD, $\mathbf{SS}^\top = \mathbf{M}^\top \mathbf{V}^\top \mathbf{NN}^\top \mathbf{VM} = \mathbf{I}$. We thus know $\mathbf{S}$ is an orthogonal matrix and the magnitude of its diagonal elements cannot exceed 1. Since $\mathbf{D}_{kk}$ are all non-negative. To maximize $\sum_{k=1}^r \mathbf{D}_{kk}\mathbf{S}_{kk}$, we let $\mathbf{S}_{kk} = 1$ for all $1 \leq k \leq r$. This is equivalent to setting $\mathbf{S} = \mathbf{M}^\top \mathbf{VN} = \mathbf{I}$. Therefore, one solution is given by $\mathbf{V} = \mathbf{MN}^\top$. The maximum value of the objective is thus $\sum_{k=1}^r \mathbf{D}_{kk} = \|\mathbf{Z}\|_*$, the nuclear norm of $\mathbf{Z}$.

Proof of Theorem 2

We notice that in Problem (2.2) we can solve explicitly for $\mathbf{V}$ and plug back into the objective function. It yields the objective function (after dropping the constant term $(1/2)\|\mathbf{Y}\|_F^2$):

$$F_\lambda(\mathbf{U}) = \frac{1}{2}\|\mathbf{XU}\|_2^2 - \|\mathbf{Y}^\top \mathbf{XU}\|_* + \lambda \sum_{j=1}^p \|\mathbf{U}_{j\cdot}\|_2,$$

We let $f_\lambda(\mathbf{U}) = (1/2)\|\mathbf{XU}\|_2^2 - \|\mathbf{Y}^\top\mathbf{XU}\|_*$ without the penalty term so that $F_\lambda(\mathbf{U}) = f_\lambda(\mathbf{U}) + \lambda\sum_{j=1}^p \|\mathbf{U}_{j\cdot}\|_2$. Define a local smooth approximation of F_λ as

$$\tilde{F}_\lambda^t(\mathbf{U}';\mathbf{U}) = f_\lambda(\mathbf{U}) + \langle\nabla f_\lambda(\mathbf{U}), \mathbf{U}'-\mathbf{U}\rangle + (1/2t)\|\mathbf{U}'-\mathbf{U}\|_F^2 + \lambda\sum_{j=1}^p \|\mathbf{U}_{j\cdot}\|_2,$$

and $\mathbf{U}^+ = \mathrm{argmin}_{\mathbf{U}'}[\tilde{F}_\lambda^t(\mathbf{U}';\mathbf{U}) - F_\lambda(\mathbf{U})]$. Dubois et al. [2019] showed that if t is small enough such that $\tilde{F}_\lambda^t(\mathbf{U}^+;\mathbf{U}) \geq F_\lambda(\mathbf{U}^+)$, we have

$$F_\lambda(\mathbf{U}^+) - F_\lambda^* \leq \left(1 - \min\left(\frac{1}{2}, \mu t\right)\right)(F_\lambda(\mathbf{U}) - F_\lambda^*). \tag{2.14}$$

Consider the iterates $(\mathbf{U}^k, \mathbf{V}^k)_{k\geq 1}$ in the alternating minimization algorithm. Notice that $\nabla f_\lambda(\mathbf{U}^k) = \mathbf{X}^\top\mathbf{XU}^k - \mathbf{X}^\top\mathbf{YV}^k$. We have

$$\begin{aligned}
F_\lambda(\mathbf{U}^{k+1}) &= g(\mathbf{U}^{k+1},\mathbf{V}^{k+1}) - \frac{1}{2}\|\mathbf{Y}\|_F^2 \quad (g \text{ is the SRRR objective function}) \\
&\leq g(\mathbf{U}^{k+1},\mathbf{V}^k) - \frac{1}{2}\|\mathbf{Y}\|_F^2 \\
&= \min_{\mathbf{U}} g(\mathbf{U},\mathbf{V}^k) - \frac{1}{2}\|\mathbf{Y}\|_F^2 \\
&= \min_{\mathbf{U}}\left(\frac{1}{2}\|\mathbf{Y}-\mathbf{XU}^k(\mathbf{V}^k)^\top\|_F^2 + \langle\mathbf{X}^\top(\mathbf{XU}^k-\mathbf{YV}^k), \mathbf{U}-\mathbf{U}^k\rangle + \right. \\
&\quad \left.\frac{1}{2}\,\mathbf{Tr}((\mathbf{U}-\mathbf{U}^k)^\top\mathbf{X}^\top\mathbf{X}(\mathbf{U}-\mathbf{U}^k))\right) + \lambda\sum_{j=1}^p\|\mathbf{U}_{j\cdot}\|_2 - \frac{1}{2}\|\mathbf{Y}\|_F^2 \\
&\leq \min_{\mathbf{U}}\left(f_\lambda(\mathbf{U}^k) + \langle\nabla f_\lambda(\mathbf{U}), \mathbf{U}'-\mathbf{U}\rangle + \frac{1}{2}\sigma_{\max}^2\|\mathbf{U}-\mathbf{U}^k\|_F^2\right) + \lambda\sum_{j=1}^p\|\mathbf{U}_{j\cdot}\|_2 \\
&= \min_{\mathbf{U}}\tilde{F}_\lambda^{1/\sigma_{\max}^2}(\mathbf{U};\mathbf{U}^k),
\end{aligned}$$

where the fourth line is the quadratic expansion of $g(\mathbf{U},\mathbf{V}^k)$ at $\mathbf{U}^k$, the second to last is by the fact that $\mathbf{Tr}((\mathbf{U}-\mathbf{U}^k)^\top\mathbf{X}^\top\mathbf{X}(\mathbf{U}-\mathbf{U}^k)) \leq \sigma_{\max}^2\|\mathbf{U}-\mathbf{U}^k\|_F^2$, and the last equality is by the definition of $\tilde{F}_\lambda^t$ function. Therefore, if we let $\mathbf{U}^{k,+} = \mathrm{argmin}_{\mathbf{U}}[\tilde{F}_\lambda^{1/\sigma_{\max}^2}(\mathbf{U};\mathbf{U}^k) - F_\lambda(\mathbf{U}^k)]$, we have

$$F_\lambda(\mathbf{U}^{k+1}) - F_\lambda^* \leq F_\lambda(\mathbf{U}^{k,+}) - F_\lambda^*. \tag{2.15}$$

We need to show that $\mathbf{U}^{k,+}$ satisfies the condition $\tilde{F}_\lambda^{1/\sigma_{\max}^2}(\mathbf{U}^{k,+};\mathbf{U}^k) \geq F_\lambda(\mathbf{U}^{k,+})$. To see this, notice that in fact for any $\mathbf{U}$,

$$\begin{aligned}\frac{1}{2}\|\mathbf{XU}\|_2^2 &= \frac{1}{2}\|\mathbf{XU}^k\|_F^2 + \langle \mathbf{X}^\top\mathbf{XU}^k, \mathbf{U}-\mathbf{U}^k\rangle + \frac{1}{2}\|\mathbf{X}(\mathbf{U}-\mathbf{U}^k)\|_F^2 \\ &\leq \frac{1}{2}\|\mathbf{XU}^k\|_F^2 + \langle \mathbf{X}^\top\mathbf{XU}^k, \mathbf{U}-\mathbf{U}^k\rangle + \frac{1}{2}\sigma_{\max}^2\|\mathbf{U}-\mathbf{U}^k\|_F^2.\end{aligned}$$

Since $\mathbf{X}^\top\mathbf{YV}^k$ is a subgradient of $\|\mathbf{Y}^\top\mathbf{XU}\|_*$ at $\mathbf{U}^k$, we have

$$-\|\mathbf{Y}^\top\mathbf{XU}\|_* \leq -\|\mathbf{Y}^\top\mathbf{XU}^k\|_* - \langle \mathbf{X}^\top\mathbf{YV}^k, \mathbf{U}-\mathbf{U}^k\rangle.$$

Adding the two inequalities up, and we have $F_\lambda(\mathbf{U}) \leq \tilde{F}_\lambda^{1/\sigma_{\max}^2}(\mathbf{U};\mathbf{U}^k)$ for all $\mathbf{U}$. In particular, it holds for $\mathbf{U}^{k+}$. Therefore, by (2.14) and (2.15), we have

$$F_\lambda(\mathbf{U}^{k+1}) - F_\lambda^* \leq F_\lambda(\mathbf{U}^{k,+}) - F_\lambda^* \leq \left(1 - \min\left(\frac{1}{2}, \frac{\mu}{\sigma_{\max}^2}\right)\right)(F_\lambda(\mathbf{U}^k) - F_\lambda^*),$$

and the convergence is linear.

2.9.2 Connection with CCA

Canonical Correlation Analysis (CCA) has an internal connection with Reduced-Rank Regression (RRR). In particular, it can be shown that the low-rank components constructed on the $\mathbf{X}$ space turn out to be the same by a relaxed CCA and a generalized RRR. CCA finds linear combinations $\mathbf{XU} \in \mathbb{R}^{n\times r}$ of variables in $\mathbf{X} \in \mathbb{R}^{n\times p}$ and linear combinations $\mathbf{YV} \in \mathbb{R}^{n\times r}$ of variables in $\mathbf{Y} \in \mathbb{R}^{n\times q}$ that attain the maximum correlation. We assume both $\mathbf{X}$ and $\mathbf{Y}$ have been centered. CCA solves the following optimization problem:

$$\begin{aligned}\underset{\mathbf{U},\mathbf{V}}{\text{maximize}} \quad & \text{tr}(\mathbf{U}^\top\mathbf{X}^\top\mathbf{YV}), \\ \text{s.t.} \quad & \mathbf{U}^\top\mathbf{X}^\top\mathbf{XU} = \mathbf{V}^\top\mathbf{Y}^\top\mathbf{YV} = \mathbf{I}_r.\end{aligned} \tag{2.16}$$

In particular, in the one dimensional case, this reduces to the problem of maximizing our familiar correlation coefficient. An equivalent representation to (2.16) can be written as

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad & \|\mathbf{YV} - \mathbf{XU}\|_F^2, \\ \text{s.t.} \quad & \mathbf{U}^\top \mathbf{X}^\top \mathbf{XU} = \mathbf{V}^\top \mathbf{Y}^\top \mathbf{YV} = \mathbf{I}_r. \end{aligned} \tag{2.17}$$

The solution to the problem is $\hat{\mathbf{U}} = \mathbf{S}_{xx}^{-1/2}\mathbf{Q}^{(r)}, \hat{\mathbf{V}} = \mathbf{S}_{yy}^{-1/2}\mathbf{P}^{(r)}$ where $\mathbf{P}^{(r)}$ and $\mathbf{Q}^{(r)}$ are the r leading left and right singular vectors of matrix $\mathbf{R} = \mathbf{S}_{yy}^{-1/2}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1/2}$. $\mathbf{P}^{(r)}$ is also the r leading eigenvectors of $\mathbf{S}_{yy}^{-1/2}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1/2}$. A relaxed form of CCA problem ignoring the $\mathbf{U}$-constraint solves

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad & \|\mathbf{YV} - \mathbf{XU}\|_F^2, \\ \text{s.t.} \quad & \mathbf{V}^\top \mathbf{Y}^\top \mathbf{YV} = \mathbf{I}_r. \end{aligned} \tag{2.18}$$

The solution is $\hat{\mathbf{U}} = \mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1/2}\mathbf{P}^{(r)}, \hat{\mathbf{V}} = \mathbf{S}_{yy}^{-1/2}\mathbf{P}^{(r)}$, where $\mathbf{P}^{(r)}$ is the r leading eigenvectors of $\mathbf{S}_{yy}^{-1/2}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1/2}$. Therefore, the solution for $\mathbf{V}$ remains unchanged, though $\mathbf{U}$ is different due to the constraint.

On the other hand, in the (generalized) reduced rank regression, given a given positive-definite matrix Γ, the problem becomes

$$\underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad \text{tr}(\Gamma^{1/2}(\mathbf{Y} - \mathbf{XUV}^\top)^\top(\mathbf{Y} - \mathbf{XUV}^\top)\Gamma^{1/2}). \tag{2.19}$$

This can be derived, for example, as an maximum likelihood estimator under the Gaussian assumption with known covariance Γ^{-1}. One solution [Velu and Reinsel, 2013] is given by

$$\begin{aligned} \hat{\mathbf{U}} &= \mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\Gamma^{1/2}\mathbf{P}^{(r)}, \\ \hat{\mathbf{V}} &= \Gamma^{-1/2}\mathbf{P}^{(r)}, \end{aligned}$$

where $\mathbf{P}^{(r)}$ is the leading eigenvectors of $\mathbf{R} = \Gamma^{1/2}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\Gamma^{1/2}$. We see that the solution when $\Gamma = \mathbf{S}_{yy}^{-1}$ is closely related to the relaxed CCA solution. $\mathbf{U}$ is the same while $\mathbf{V}$ is the so-called reflexive inverse of $\mathbf{V}$ there.

2.9.3 Additional Experiments

We conduct some experiments to gain more insight into the method and compare with other methods. We generate the $\mathbf{X} \in \mathbb{R}^{n\times p}$ with independent samples from some multivariate Gaussian $\mathcal{N}(0, \Sigma_X)$. For the first several cases, we generate the response from the true, most favorable model $\mathbf{Y} = \mathbf{XUV}^\top + \mathbf{E}$, where each entry in the support of $\mathbf{U} \in \mathbb{R}^{p\times r}$ (sparsity k) is independently drawn from a standard Gaussian distribution, and $\mathbf{V} \in \mathbb{R}^{q\times r}$ takes the left singular matrix of a Gaussian ensemble. Hence $\mathbf{B} = \mathbf{UV}^T$ is the true coefficient matrix. The noise matrix is generated from $\mathcal{N}(0, \sigma_e^2\Sigma_E)$, where σ_e^2 is chosen such that the signal-to-noise ratio

$$\text{SNR} = \frac{\text{tr}(\mathbf{B}^\top\Sigma_X\mathbf{B})}{\sigma_e^2 \cdot \text{tr}(\Sigma_E)} \tag{2.20}$$

is set to a given level. The performance is evaluated by the test R^2, defined as follows:

$$R^2 = 1 - \frac{\|\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}\|_F^2}{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_F^2}.$$

We consider several sets of experiments.

1. **Scenario 1-9** Small experiments: $(n, p, k) = (200, 100, 20), (200, 500, 20), (200, 500, 50), q = 20, r = 3$. The X has independent design, and the noise across different responses are all independent, i.e. $\Sigma_X = \mathbf{I}_p, \Sigma_E = \mathbf{I}_q$. Target SNR = 0.5, 1, 3. The results are evaluated on test sets of size 5000.

2. **Scenario 10-18** Same as Scenario 1-9. The true coefficient matrix is no longer exact low rank. It is perturbed by Gaussian noise with mean 0 and standard deviation 0.5.

3. **Scenario 19-27** Same as Scenario 1-9, except that the predictors are correlated. In particular,

$$\text{Cov}\left[(\right] \mathbf{x}_j, \mathbf{x}_{j'}) = \begin{cases} 1, & j = j', \\ \rho, & j \neq j'. \end{cases}$$

   We let $\rho = 0.5$ in this set of simulation.

4. **Scenario 28-36** Same as Scenario 10-18, except that the predictors are correlated as in Scenario 19-27.

From the simulations, we find that underestimating the rank can degrade the performance instantly. Overestimating the rank will give one a variance penalty, but it seems to be rather robust compared with the other direction.

Scenario 1-9 Small experiments: $(n, p, k) = (200, 100, 20), (200, 500, 20), (200, 500, 50), q = 20, r = 3$. The $\mathbf{X}$ has independent design, and the noise across different responses are all independent, i.e. $\Sigma_X = \mathbf{I}_p, \Sigma_E = \mathbf{I}_q$. Target SNR = 0.5, 1, 3. The results are evaluated on test sets of size 5000.

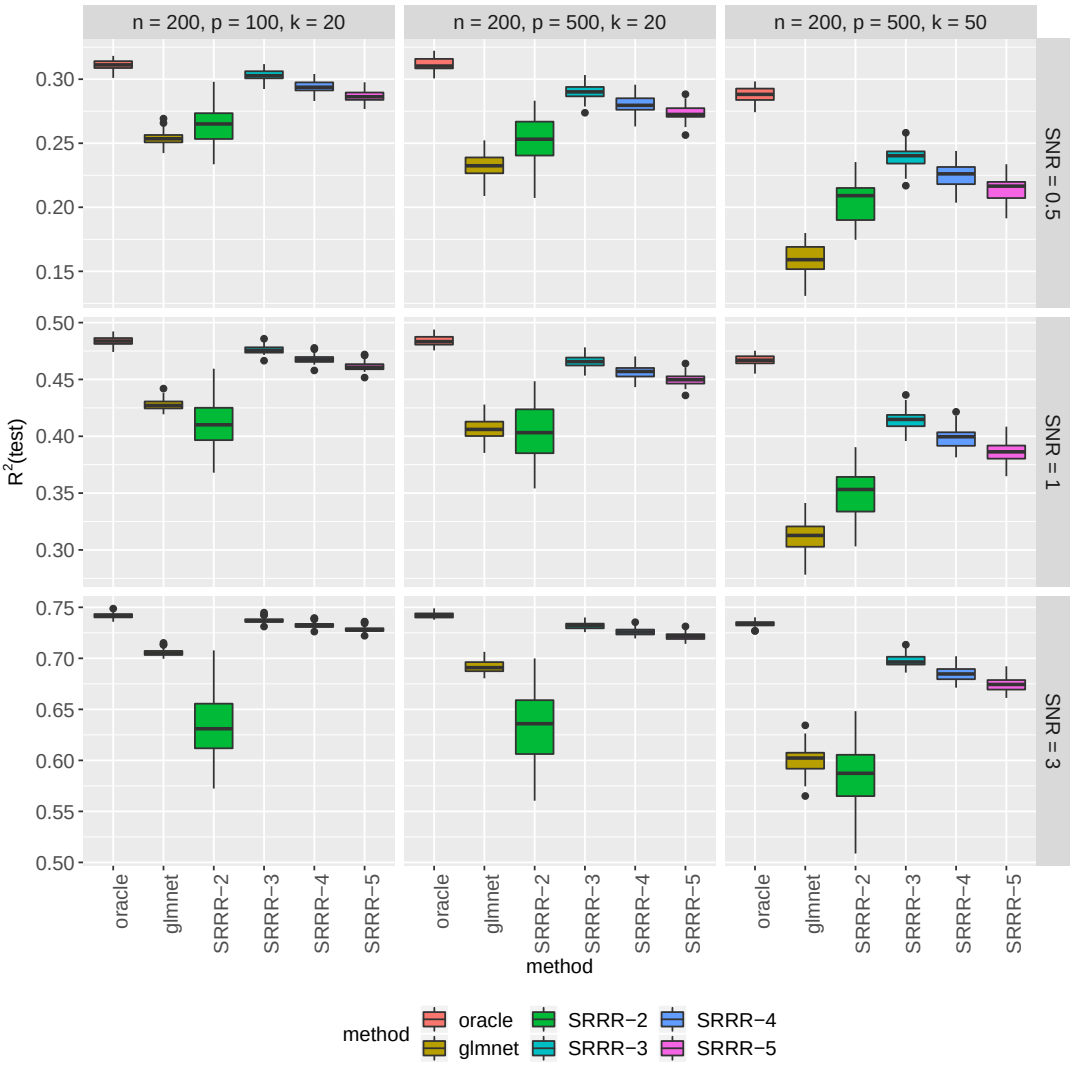


Figure 2.9: Scenario 1-9. R^2 each run is evaluated on a test set of size 5000.

Scenario 10-18 Same as Scenario 1-9. The true coefficient matrix is no longer exact low rank. It is perturbed by Gaussian noise with mean 0 and standard deviation 0.5.

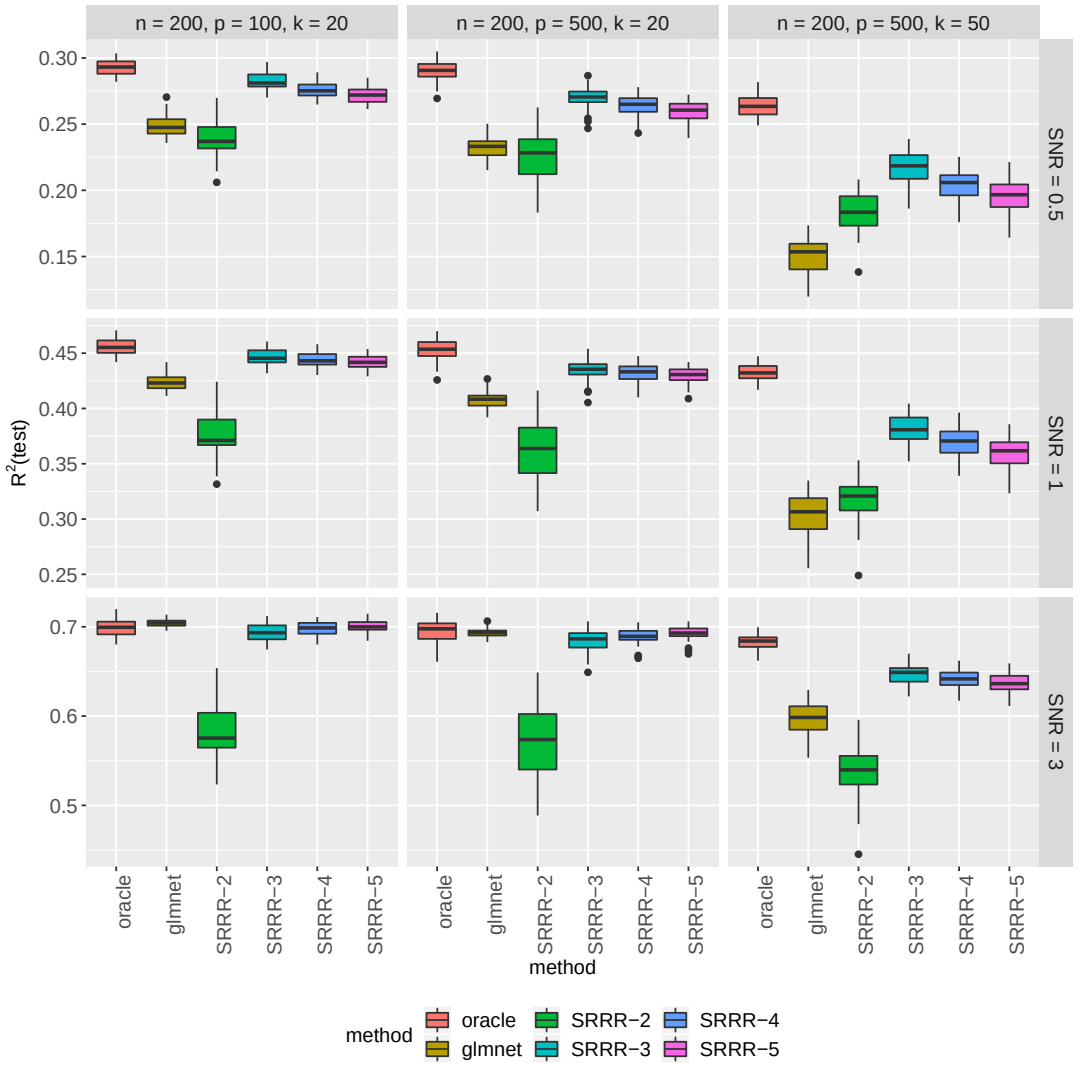


Figure 2.10: Scenario 10-18. R^2 each run is evaluated on a test set of size 5000. The oracle here does not take into account the noise in true coefficient matrix, and do reduced rank regression on the true support and the true rank.

Scenario 19-27 Same as Scenario 1-9, except that the predictors are correlated. In particular,

$$\text{Cov}\left[(\right] \mathbf{x}_j, \mathbf{x}_{j'}) = \begin{cases} 1, & j = j', \\ \rho, & j \neq j'. \end{cases}$$

We let $\rho = 0.5$ in this set of simulation.

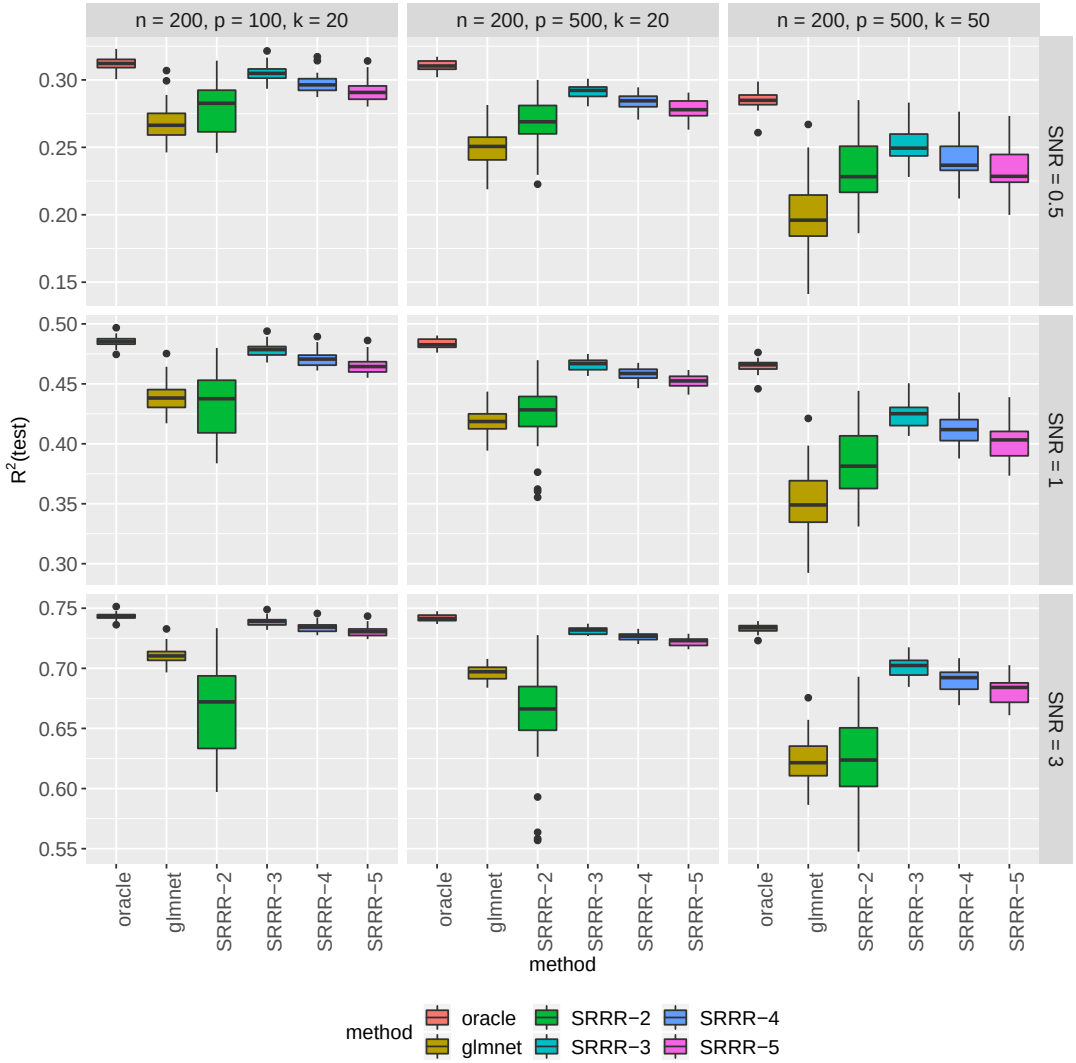


Figure 2.11: Scenario 19-27. R^2 each run is evaluated on a test set of size 5000.

Scenario 28-36 Same as Scenario 10-18, except that the predictors are correlated as in Scenario 19-27.

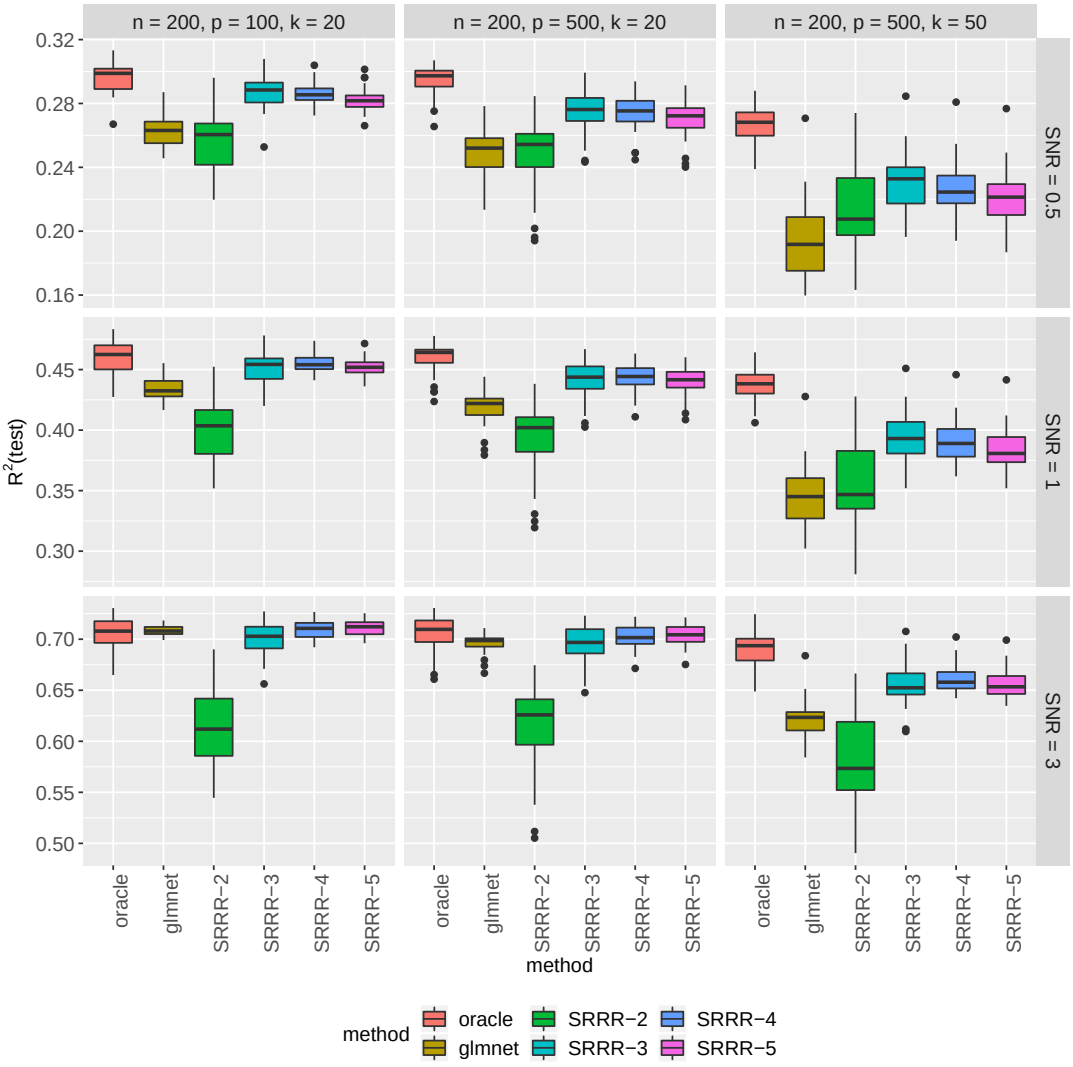


Figure 2.12: Scenario 28-36. R^2 each run is evaluated on a test set of size 5000.

2.9.4 Additional Information on the Methods

Compliance with ethical regulations and informed consent

This research has been conducted using the UK Biobank Resource under Application Number 24983, "Generating effective therapeutic hypotheses from genomic and hospital linkage data" (`http://www.ukbiobank.ac.uk/wp-content/uploads/2017/06/24983-Dr-Manuel-Rivas.pdf`). Based on the information provided in Protocol 44532 the Stanford IRB has determined that the research does not involve human subjects as defined in 45 CFR 46.102(f) or 21 CFR 50.3(g). All participants of UK Biobank provided written informed consent (more information is available at `https://www.ukbiobank.ac.uk/2018/02/gdpr/`).

Population stratification in UK Biobank

We used genotype data from the UK Biobank dataset release version 2 and the hg19 human genome reference for all analyses in the study. To minimize the variabilities due to population structure in our dataset, we restricted our analyses to include 337,151 White British individuals (Figure 2.13) based on the following five criteria [DeBoever et al., 2018, Tanigawa et al., 2019] reported by the UK Biobank in the file "ukb_sqc_v2.txt":

1. self- reported white British ancestry ("in_white_British_ancestry_subset" column)
2. used to compute principal components ("used_in_pca_calculation" column)
3. not marked as outliers for heterozygosity and missing rates ("het_missing_outliers" column)
4. do not show putative sex chromosome aneuploidy ("putative_sex_chromo- some_aneuploidy" column)
5. have at most 10 putative third-degree relatives ("excess_relatives" column).

Variant annotation and quality control

We prepared a genotype dataset by combining the directly-genotype variants, copy number variants (CNVs) and HLA allelotype datasets.

We annotated the directly-genotyped variants using the VEP LOFTEE plugin (`https://github.com/konradjk/loftee`) and variant quality control by comparing allele frequencies in the UK

Biobank and gnomAD (gnomad.exomes.r2.0.1.sites.vcf.gz) as previously described28. We focused on variants outside of the major histocompatibility complex (MHC) region (chr6:25477797-36448354) as previously described. We focused on the variants according to the following criteria:

- Missigness of the variant is less than 1%, considering that two genotyping arrays (the UK BiLEVE array and the UK Biobank array) which covers a slightly different set of variants.
- Minor-allele frequency is greater than 0.01%, given the recent reports casting questions on the reliability of ultra low-frequency variants.
- The variant is in the LD-pruned set
- Hardy-Weinberg disequilibrium test p-value is less than 1.0×10^{-7}
- Manual cluster plot inspection. We investigated the cluster plots for subset of variants and removed 11 variants that have unreliable genotype calls.
- Passed the comparison of minor allele frequency with gnomAD dataset as described before

CNVs were called by applying PennCNV v1.0.4 on raw signal intensity data from each array within each genotyping batch as previously described. We applied a filter on minor-allele frequency (MAF $>$ 0.01%), which resulted in 8,274 non-rare (MAF $>$ 0.01%) CNVs.

The HLA data from the UK Biobank contains all HLA loci (one line per person) in a specific order (A, B, C, DRB5, DRB4, DRB3, DRB1, DQB1, DQA1, DPB1, DPA1). We downloaded these values, which were imputed via the HLA:IMP*2 program (Resource 182); the UK Biobank reports one value per imputed allele, and only the best-guess alleles are reported. Out of the 362 alleles reported in UKB, we used 175 alleles that were present in >0.1% of the population surveyed.

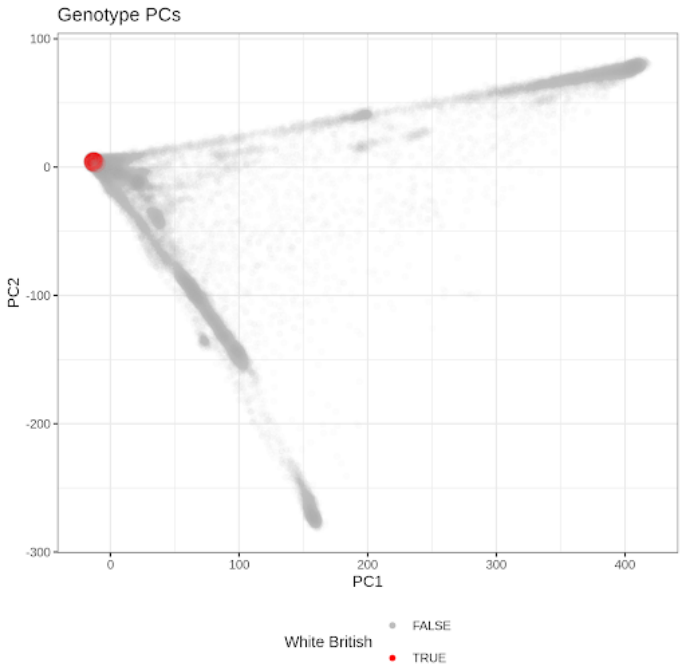


Figure 2.13: The identification of unrelated White British individuals in UK Biobank. The first two genotype principal components (PCs) are shown on the x- and y-axis and the identified unrelated White British individuals (Methods) are shown in red.

Chapter 3

Some Methods for Heterogeneous Treatment Effect Estimation in High Dimensions

3.1 Introduction

In February 2017, at the Grand Rounds of Stanford Medicine, one of us (NS) unveiled a new initiative — the Informatics Consult. Through this service, clinicians can submit a consultation request online and receive a report based on insights drawn from hundreds of millions of electronic medical records (EMRs) from Stanford Health Care. While the system is in its early stages, a future version will include treatment recommendations: helping a doctor to choose between treatment options for a patient, in cases where there is no randomized controlled trial (RCT) which compares the options. This announcement was met with excitement from the doctors in attendance, considering that they generally need to make decisions without any support from quantitative evidence (about 95% of the time) [Shah, 2016]. Building such a system is a priority in many medical centers in the U.S. and around the world.

The problem setting on which this paper focuses is when a doctor is presented with a patient who has some medical ailment, and the doctor is considering one or more treatment options. A

relevant question from the patient's perspective is, *what is the effect of these treatments on patients like me?* Devising a meaningful definition for "patients like me" is especially difficult given the high-dimensional nature of the problem: We observe thousands of features describing each patients, any of which could be used to describe patient similarity. The other significant complication is that our goal is to infer causal effects from observational data. The task of mining EMRs to support physician decision-making is what motivates this paper. We propose and study methods for estimation and inference of heterogeneous treatment effects, for both randomized experiments and observational studies. We focus on the case of a choice between two treatments, which for the purposes of this manuscript we label as "treatment" and "control".

In detail, we have an $n \times p$ matrix of features $\mathbf{X}$, a treatment indicator vector $\mathbf{T} \in \{0,1\}^n$, and a vector of quantitative responses $\mathbf{Y} \in \mathbb{R}^n$. Let X_i denote the ith row of $\mathbf{X}$, likewise T_i and Y_i. We assume the n observations (X_i, T_i, Y_i) are sampled i.i.d. from some unknown distribution. The number of treated patients is $N_1 = |\{i : T_i = 1\}|$, and the number of control patients is $N_0 = |\{i : T_i = 0\}|$. We adopt the Neyman–Rubin potential outcomes model [Splawa-Neyman et al., 1990, Rubin, 1974]: each patient i has potential outcomes $Y_i^{(1)}$ and $Y_i^{(0)}$, only one of which is observed. $Y_i^{(1)}$ is the response that the patient would have under treatment, and $Y_i^{(0)}$ is the response the patient would have under control. Hence the outcome that we actually observe is $Y_i = Y_i^{(T_i)}$. We consider both randomized controlled trials, where T_i is independent of all pre-treatment characteristics,

$$\left(X_i,\, Y_i^{(0)},\, Y_i^{(1)}\right) \perp\!\!\!\perp T_i, \tag{3.1}$$

and observational studies, where the distribution of T_i is dependent on the covariates. This scenario is discussed in further detail in Section 3.2.1.

We describe four important functions for modelling data of this type. The first is the propensity function, which gives the probability of treatment assignment, conditional on covariates:

$$\pi(x) \equiv \mathbb{P}(T = 1 | X = x). \tag{3.2}$$

The next two functions are the conditional mean functions: the expected response given treatment and the expected response given control.

$$\mu_1(x) \equiv \mathbb{E}[Y | X = x, T = 1] \quad \text{and} \quad \mu_2(x) \equiv \mathbb{E}[Y | X = x, T = 0].$$

The fourth function, and the one of greatest interest, is the treatment effect function, which is the difference between the two conditional means:

$$\tau(x) \equiv \mu_1(x) - \mu_0(x).$$

We seek regions in predictor space where the treatment effect is relatively large or relatively small. This is particularly important for the area of personalized medicine, where a treatment might have a negligible effect when averaged over all patients but could be beneficial for certain patient subgroups.

An outline of this paper is as follows. Section 3.2 reviews related work. In Section 3.3 we describe the two main high-level approaches to the estimation of heterogeneous treatment effects: transformed outcome regression and conditional mean regression. In Section 3.4 we introduce *pollinated transformed outcome* (PTO) forests, while *causal boosting* is proposed in Section 3.5. *Causal MARS* is the focus of Section 3.6. In Section 3.7 we report the results of a simulation study comparing all of these methods, and a real data application is illustrated in Section 3.8. We end with a discussion.

3.2 Related work

Early work on heterogeneous treatment effect estimation [Gail and Simon, 1985] was based on comparing pre-defined subpopulations of patients in randomized experiments. To characterize interactions between a treatment and continuous covariates, Bonetti and Gelber [2004] formalized the subpopulation treatment effect patter plot (STEPP). Sauerbrei et al. [2007] proposed an efficient algorithm for flexible model-building with multivariable fractional polynomial interaction (MFPI) and compared the empirical performance of MFPI with STEPP.

Identifying subgroups within the patient population is becoming especially problematic in high-dimensional data, as in EMRs. In recent years, a great amount of work has been done to apply methods from machine learning to let the data inform what are the important subgroups in terms of treatment effect. Su et al. [2009] proposed interaction trees for adaptively defining subgroups based on treatment effect. Athey and Imbens [2016] proposed causal trees, which are similar, and constructed valid confidence intervals. Wager and Athey [2015] improved on this line of work by growing

random forests [Breiman, 2001] from causal trees. These tree-based methods all use shared-basis conditional mean regression in the framework of Section 3.3. An example of a transformed-outcome estimator is the FindIt method of Imai and Ratkovic [2013] which trains an adapted support vector machine on a transformed binary outcome. Tian et al. [2014] introduced a simple linear model based on transformed covariates and show that it is equivalent to transformed outcome regression in the Gaussian case. In a novel approach, Zhao et al. [2012] used outcome weighted learning to directly determine individualized treatment rules, skipping the step of estimating individualized treatment effects. The problem of estimating heterogeneous treatment effects has also received significant attention in Bayesian literature. Hill [2011] and Green and Kern [2012] approached the problem using Bayesian additive regression trees [Chipman et al., 1998], and Taddy et al. [2016] proposed a method based on Bayesian forests. Chen et al. [2012] developed a Bayesian method for finding qualitative interactions between treatment and covariates, and there are other Bayesian methods for flexible nonlinear modelling of interactive/non-additive relationships between covariates and response [LeBlanc, 1995, Gustafson, 2000].

What all of the above work (except Hill [2011]) have in common is that they assume randomized treatment assignment. Athey and Imbens [2016] discussed the possibility of adapting their method to observational data but go no further. Wager and Athey [2015] proposed the propensity forest when treatment is not randomized, but this method does not target heterogeneity in the treatment effect. Similarly, Xie et al. [2012] model treatment effect as a function of propensity score, missing out on how it depends on the covariates except through treatment propensity. Crump et al. [2008] devised a nonparametric test for the null hypothesis that the treatment effect is constant across patients, but that is not suited to high-dimensional data. One promising approach which flexibly handles high-dimensional and observational data is the gradient forest of Athey et al. [2017]—we compare the performance of our methods with that of the gradient forest in Section 3.7.

We are particularly interested in flexible, non-parametric approaches that can handle large numbers of observations and predictors, and model interactions between predictors, which none of these papers deal with (except for Zhao et al. [2012]).

3.2.1 Propensity score methods

Much of causal inference is based on the propensity score [Rosenbaum and Rubin, 1983], which is the estimated probability that a patient would receive treatment, conditioned on the patient's

covariates. If the estimate of the propensity function (3.2) is $\hat{\pi}(\cdot)$, then the propensity score for a patient with covariate vector x is $\hat{\pi}(x)$. Throughout the present work, we estimate the propensity function using the probability forests of Malley et al. [2012]. We are able to do so quickly using the fast implement in the R package `ranger` [Wright and Ziegler, 2017].

For the estimation of a population-average treatment effect (ATE), propensity score methods for reducing bias in observational studies have been established [Austin, 2011]. *Propensity score matching* emulates a randomized control trial (RCT) by choosing pairs of patients with similar propensity scores, one each in the treatment and control arms, and discards the unmatched patients. *Stratification on the propensity score* groups patients into bins of similar propensity scores to compute the ATE within each bin. The overall ATE is the average of these treatment effects, weighted by the overall frequency of each bin. *Inverse probability weighting* assigns a weight to each patient equal to the inverse of the propensity score if the patient is treated, or else the inverse of one minus the propensity score if the patient is not treated. Hence patients who tend to be under-represented in their arm are given more weight. Propensity score stratification and inverse probability weighting are discussed in more detail in the appendix, along with an additional method: *transformed outcome averaging*.

The assumption that enables these methods to generate causal conclusions from observational data is known alternatingly across the literature as unconfoundedness, exogeneity or strong ignorability:

$$(Y_i^{(1)}, Y_i^{(0)}) \perp\!\!\!\perp T_i | X_i$$

This is the assumption made in the present work. It means that the relationship between the potential outcomes and treatment must be fully explained by X. There can be no additional unmeasured confounding variable which effects a dependence between potential outcomes and treatment. Note, however, that the outcome itself is not independent of treatment because the treatment determines which potential outcome is observed.

Low et al. [2016] cast doubt on the ability of propensity score methods to adequately account for selection bias in a sophisticated simulation designed to model reality. Nevertheless, we observe in Section 3.7 that propensity score adjustments improve results in non-randomized simulations, which means that they can be used to help doctors make more informed decisions, so we push forward with the application of propensity scores.

3.3 Transformed outcome regression and conditional mean regression

Methods for estimating heterogeneous treatment effects generally fall into one of two categories: *transformed outcome regression* or *conditional mean regression*. In this section we describe the two approaches and explain why we prefer conditional mean regression. The propensity transformed outcome method (Section 3.4) uses a combination of the two approaches, while causal forests (Section 3.2), causal boosting (Section 3.5), and causal MARS (Section 3.6) are all conditional mean regression methods.

Transformed outcome regression is based on the same idea as transformed outcome averaging, which is laid out in detail in the appendix. Given the data described in Section 3.1, we define the *transformed outcome* as

$$Z \equiv T\frac{Y}{\pi(X)} + (1-T)\frac{-Y}{1-\pi(X)}.$$

This quantity is interesting because, as shown in the appendix, for any covariate vector x, $\mathbb{E}[Z|X = x] = \tau(x)$. So the transformed outcome gives us for each patient an unbiased estimate of the personalized treatment effect for that patient. Using this, we can simply use the tools of supervised learning to estimate a regression function for the mean of Z given X. The weakness of this approach is that while Z is unbiased for the treatment effect, its variance can be large due the presence of the propensity score, which can be close to zero or one, in the denominator.

An alternative approach, conditional mean regression is based on the idea that because $\tau(x)$ is defined as the difference between $\mu_1(x)$ and $\mu_0(x)$, if we can get good estimates of these conditional mean functions, then we have a good estimate of the treatment effect function. Estimating the functions $\mu_1(x)$ and $\mu_0(x)$ are supervised learning problems. If they are both estimated perfectly, then there is no need to bother with propensity scores. The problem is that in practice we never estimate either function perfectly, and differences between the covariate distributions in the two treatment groups can lead to bias in treatment effect estimation if propensity scores are ignored.

We compare these two approaches with a simple example: Consider the task of estimating an ATE using data from a randomized trial. This may seem far removed from heterogeneous treatment effect estimation, but we will describe how two of our methods are based on estimating local ATEs

for subpopulations in our data. In this case, the transformed outcome is

$$Z = T\frac{Y}{1/2} + (1-T)\frac{-Y}{1/2} = 2TY - 2(1-T)Y,$$

and the corresponding estimate of the ATE is

$$\hat{\tau}_{\text{TO}} = \frac{1}{n}\sum_{i=1}^{n} Z_i = \frac{2N_1\bar{Y}_1 - 2N_0\bar{Y}_0}{N_1+N_0} = \frac{N_1}{n/2}\bar{Y}_1 - \frac{N_0}{n/2}\bar{Y}_0,$$

where $\bar{Y}_1$ is the average response of patients who received treatment and $\bar{Y}_0$ is the average response of control patients. Meanwhile the conditional mean estimator of the ATE would be

$$\hat{\tau}_{\text{CM}} = \bar{Y}_1 - \bar{Y}_0.$$

Here we are implicitly assuming that neither N_1 nor N_0 is zero. It is worth noting that

$$\hat{\tau}_{\text{TO}} = \hat{\tau}_{\text{CM}} + \frac{N_1 - N_0}{n}(\bar{Y}_1 + \bar{Y}_0),$$

so if $N_1 = N_0$ or $\bar{Y}_1 + \bar{Y}_0 = 0$, then $\hat{\tau}_{\text{TO}} = \hat{\tau}_{\text{CM}}$. However N_1, N_0, $\bar{Y}_1$ and $\bar{Y}_0$ are all random. Given a fixed sample size n, N_1 follows a Binomial$(n, 1/2)$ distribution (truncated to exclude 0 and n), and N_0 is the difference between n and N_1. Suppose $\bar{Y}_1$ and $\bar{Y}_0$ have normal distributions with variances inversely proportional to sample size:

$$\bar{Y}_1 \sim \text{Normal}(\mu_1, \sigma^2/N_1) \qquad \text{and} \qquad \bar{Y}_0 \sim \text{Normal}(\mu_0, \sigma^2/N_0).$$

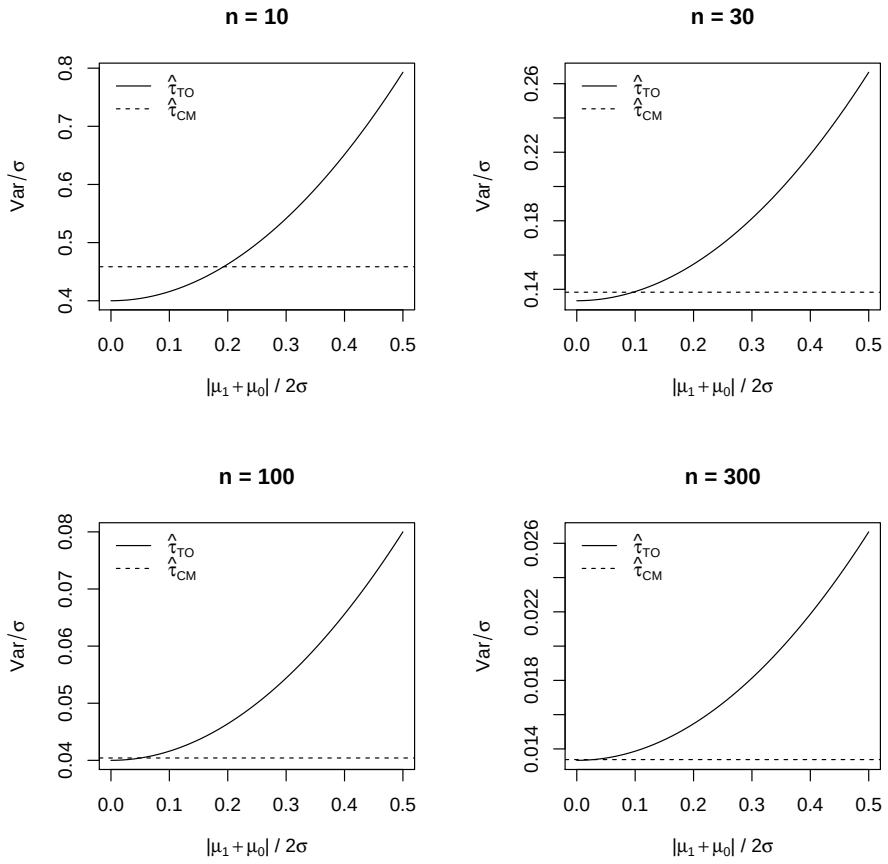
Note that both $\hat{\tau}_{\text{CM}}$ and $\hat{\tau}_{\text{TO}}$ are unbiased for $\tau \equiv \mu_1 - \mu_0$, but the two estimators have different variances. Conditioning on N_1, the variance of $\hat{\tau}_{\text{CM}}$ is

$$\mathbb{E}[(\hat{\tau}_{\text{CM}} - \tau)^2 | N_1] = \mathbb{V}(\bar{Y}_1 - \bar{Y}_0 | N_1) = \sigma^2/N_1 + \sigma^2/N_0$$

while the variance of $\hat{\tau}_{\text{TO}}$ given N_1 is

$$\mathbb{E}[(\hat{\tau}_{\text{TO}} - \tau)^2 | N_1] = \mathbb{V}(\hat{\tau}_{\text{TO}} | N_1) + (\mathbb{E}[\hat{\tau}_{\text{TO}} - \tau | N_1])^2 = \frac{4}{n}\sigma^2 + \left(\frac{N_1 - N_0}{n}\right)^2 (\mu_1 + \mu_0)^2.$$

Figure 3.1: The variance of two ATE estimators for $n = 10$, 30, 100 and 300, as the ratio of the absolute main effect $|\mu_1 + \mu_0|/2$ to the noise level σ increases from 0 to 0.5.



So the key is the ratio of the main effect $(\mu_1 + \mu_0)/2$ to the noise level σ. If

$$\left|\frac{\mu_1 + \mu_0}{2\sigma}\right| < \sqrt{\frac{N_1^{-1} + N_0^{-1} - 4n^{-1}}{(N_1 - N_0)^2}},$$

then $\hat{\tau}_{\mathrm{TO}}$ has less variance. If the inequality is reversed, then $\hat{\tau}_{\mathrm{CM}}$ has less variance. Marginalizing over the truncated binomial distribution of N_1 is difficult to do analytically, but we can numerically estimate the marginal variance of each estimator for any $n > 1$. Figure 3.1 illustrates the results for a few different choices of n.

We observe that for small n, $\hat{\tau}_{\mathrm{TO}}$ can have slightly smaller variance than $\hat{\tau}_{\mathrm{CM}}$ if the absolute

value of the main effect is close to zero. But this advantage tends to zero as n increases, and $\hat{\tau}_{\text{TO}}$ has much greater variance if the main effect is large. In conclusion, we prefer the conditional mean estimator because of the potentially high variance of the transformed outcome estimator. This is reflected in the following sections as all of our methods use some version of conditional mean regression.

3.3.1 Shared-basis conditional mean regression

In high-dimensional data it is often necessary to choose a subset of variables to include in a model. Beyond that, nonparametric methods adaptively choose transformations of variables. Collectively, we refer to the variables and transformations selected as the basis of the regression. In conditional mean regression it is to be expected that the selected basis be different between the two regression functions. This can cause differences between the conditional means attributable not to evidence of a heterogeneous treatment effect but rather due to chance in basis selection.

To address this all of our methods jointly choose the same basis for both conditional mean regressions. In detail, this shared basis is chosen adaptively to best explain heterogeneity in the treatment effect, rather than explaining the variance in either treatment group. How exactly this shared basis is determined is different for each method.

3.4 Pollinated transformed outcome (PTO) forests

We first present the idea of a pollinated transformed outcome (PTO) forest in detail and then explain the various components.

In step 1 we compute an unbiased point estimate of the treatment effect for each individual; then in step 2, we fit a random forest using this effect as the outcome. In principal, this should estimate our personalized treatment effect. However, we don't trust these estimates too much, because the outcome can be highly variable. But we will put faith in the trees they produced.

Thus in step 3, we "pollinate" the trees separately with the treated and untreated populations. That is, we send data down each tree and compute new predictions for each terminal node. In step 4, the difference $z_i = G_1(x_i) - G_0(x_i)$ gives us an estimate of the treatment effect. Finally in step 5, we then post-process these predictions by fitting one more forest, primarily for interpretation.

Algorithm 1: PTO forest

1. Build a depth-controlled propensity random (regression) forest $\hat{\pi}$ using the treatment indicator as the response. Use regression trees, so that we estimate the probability of the terminal-node means. If the data are known to have come from a randomized trial, do not build a random forest and instead define $\hat{\pi}$ to be identically equal to the probability of treatment assignment.

2. Define the transformed outcome by

$$Z_i = T_1 \frac{Y_i}{\hat{\pi}(X_i)} + (1 - T_i) \frac{-Y_i}{\hat{\pi}(X_i)}.$$

3. For the randomized treatment setting, define the transformed outcome by

$$\delta_i = (2T_i - 1)Y_i.$$

   Note that if $\hat{\pi}(X_i)$ is the true probability of receiving treatment given covariates X_i, then $E[Z_i|T_i, X_i] = \tau(X_i)$, the true conditional treatment effect (see appendix for details).

4. Grow a depth-controlled random forest G_{TOF} to δ_i.

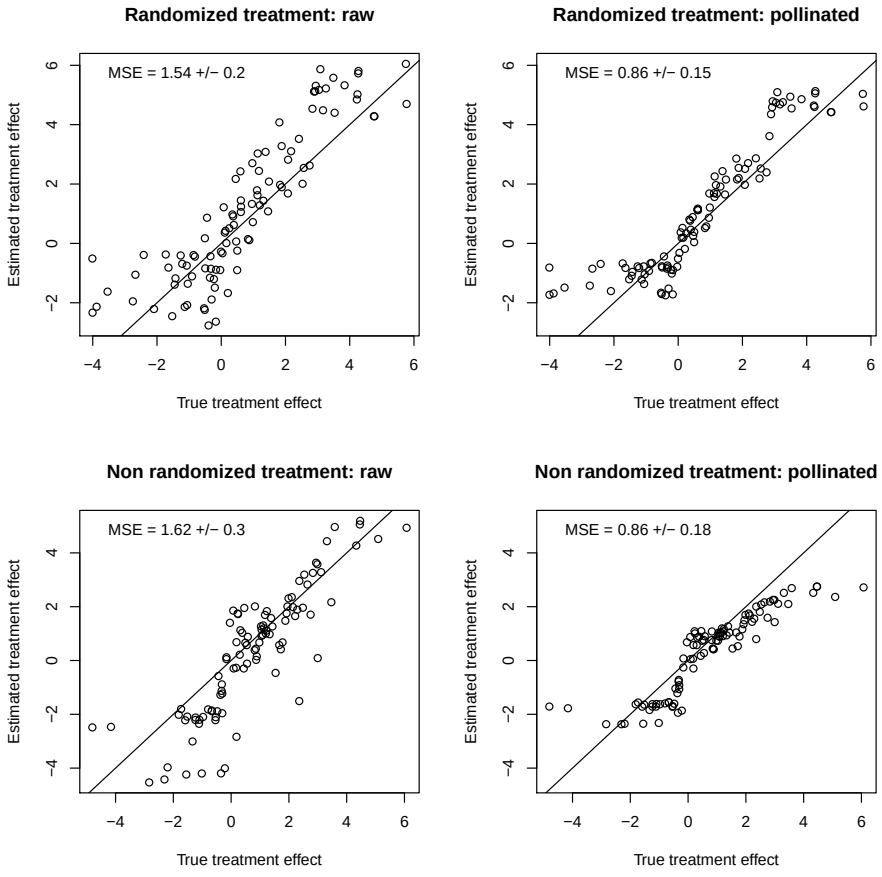
5. Pollinate G_{TOF} separately with the data in the treated group and the control group to produce two regression forests G_1 and G_0, respectively. This entails sending each observation in the treatment group down each tree in the forest to determine its terminal node and re-estimating the response in that node to be the average of its observations. The same is done for the control group.

6. Compute $\delta_i = G_1(X_i) - G_0(X_i)$.

7. Optionally, fit a random forest S to δ_i and return S, which predicts the treatment effect $\hat{\tau}(x) = S(x)$. This optional layer of regression also helps with the interpretability of the results, yielding importance scores for variables as they relate directly to the estimated treatment effect.

Figure 3.2 illustrates the benefits of cross-pollination. In this example $n = 100, p = 50$ and the response is simulated in each arm according to $Y_i \sim \mathcal{N}(1 - X_{i1} + X_{i2}, 1)$ for treated patients and $Y_i \sim \mathcal{N}(X_{i1} + X_{i2}, 1)$ for untreated patients. Hence the true personalized treatment effect for patient i is $1 + 2X_{i1}$. In the top row the treatment is randomly assigned, while in the bottom row, the probability of treatment assignment is $(1 + e^{X_{i1}+X_{i2}})^{-1}$. The raw estimates correspond to a random forest (as in step 2) grown to predict the transformed outcome. The pollinated estimates correspond to re-estimating (as in step 3) the means of the leaves within each arm. We observe that in each case, the pollination improves the estimates.

Figure 3.2: A comparison of raw and pollinated transformed outcome forests. Each method is applied to a randomized simulation and a non-randomized simulation, and we visually compare the estimated treatment effect with the true treatment effect We see that in each case, the pollination improves the estimates.



3.5 Causal boosting

An alternative to a random forest for least squares regression is boosted trees. Boosting builds up a function approximation by successively fitting weak learners to the residuals of the model at each step. In this section we generalize least squares boosting for regression [Friedman, 2001] to the problem of heterogeneous treatment effect estimation.

Given data of the form $(X_i, Y_i), i = 1, ..., n$, least squares boosting starts with a regression function $\hat{F}(x) = 0$ and residuals $R_i = Y_i - \hat{F}(x_i)$. We fit a regression tree to R_i, yielding predictions $\hat{f}_1(x)$. Then we update $\hat{F}(x) \leftarrow \hat{F}(x) + \epsilon \cdot \hat{f}_1(x)$, and $R_i \leftarrow R_i - \epsilon \cdot \hat{f}_1(x_i)$ and repeat this (say) a

few hundred times. The final prediction is simply $\hat{F}(x)$, a sum of trees shrunk by ϵ.

For our current problem, our data has the form $(X_i, T_i, Y_i), i = 1, ..., n$ with $T_i \in \{0, 1\}$. For now assume randomized treatment assignment. In the next subsection we show to handle the non-randomized case. Here is how we propose to generalize least squares boosting. As with causal forests Wager and Athey [2015], our building block is a causal tree, which returns a function $\hat{g}(x, t)$. The estimated causal effect for an observation $X = x$ is $\hat{\tau}(x) = \hat{g}(x, 1) - \hat{g}(x, 0)$. This is a standard causal tree, except that for each terminal node, we return the pair of treatment-specific means rather than the treatment effect. In other words, if observation $X_i = x$ gets you into terminal node k, where the pair of estimated means are $\hat{\mu}_{1k}$ (treated) and $\hat{\mu}_{0k}$ (untreated), then these are the values returned, respectively, for $\hat{g}(x, 1)$ and $\hat{g}(x, 0)$. The algorithm is summarized in Algorithm 2 below.

Algorithm 2: Causal Boosting

1. Set the outcome $R_i = Y_i$, and define $\hat{G}_0(x, t) = 0$.
2. Do $k = 1, ..., K$
   (a) Fit a causal tree $\hat{g}_k$ to data (X_i, R_i, T_i).
   (b) Set

$$\begin{aligned} R_i &\leftarrow R_i - \epsilon \cdot \hat{g}_k(X_i, T_i) \\ \hat{G}_k &\leftarrow \hat{G}_{k-1} + \epsilon \cdot \hat{g}_k. \end{aligned}$$

3. Return $\hat{G}_K(x, T)$.

The estimated treatment effect for any observation x is $\hat{G}_K(x, 1) - \hat{G}_K(x, 0)$.

Note that this generalizes to loss functions other than squared error. For example, if the causal tree was trained for a binary outcome, then each terminal node would return a pair of logits $\hat{\eta}_{1k} = \text{logit}[\Pr(Y = 1|X = x, T = 1)]$ and $\hat{\eta}_{0k} = \text{logit}[\Pr(Y = 1|X = x, T = 0)]$. Thus $\hat{G}_K(x, T)$ would be a function that returned a pair of logits at x, and hence treatment success probabilities. The treatment effect would be the appropriate function of these (difference, log-odds ratio). Other enhancements to boosting, such as stochastic boosting, are also applicable in the setting.

Note that causal boosting is not strictly a gradient boosting algorithm, because there is no loss function for which we are evaluating the gradient at each step, in order to minimize this loss. Rather, causal boosting is an adaptation of gradient boosting on the observed response, with a

different function in each arm of the data. The adaptation is that we use causal trees as our weak learners instead of a standard regression technique. This tweak encourages the learned function to find treatment effect heterogeneities.

3.5.1 Cross-validation for causal boosting

Unlike random forests, gradient boosting algorithms can over-fit the training data as the number of trees increases [Hastie et al., 2009]. This is because each successive tree is not built independently of the previous ones but rather with the goal of fitting to the residuals of the previous trees. Whereas a random forest will only benefit from using more trees, the number of trees in gradient boosting is itself an important parameter which needs to be tuned.

Complicating matters, the usual cross-validation framework does not apply to the setting of estimating a heterogeneous treatment effect because in this setting each observation does not come with a response corresponding directly to the function we are interested in estimating. We don't observe a response τ_i for the i^{th} patient. What we observe is either $Y_i^{(0)}$ or $Y_i^{(1)}$, depending on whether or not the patient received the treatment.

We describe our approach in the context of a held-out validation set, but this fully specifies our cross-validation procedure. Cross-validation is simply validation done by partitioning the training set into several folds and averaging the results obtained by holding out each fold as a validation set and training on all other folds. The data in this context are a training set $(\mathbf{X}^{tr}, \mathbf{T}^{tr}, \mathbf{Y}^{tr})$ and a validation set $(\mathbf{X}^{v}, \mathbf{T}^{v}, \mathbf{Y}^{v})$. After training causal boosting on $(\mathbf{X}^{tr}, \mathbf{T}^{tr}, \mathbf{Y}^{tr})$, we are left with a sequence of models $G_1(x, T), ..., G_K(x, T)$, and we would like to evaluate the performance of each of these.

To validate the performance of each of these models, we use a pollination of the causal boosting model much like step 3 of the PTO forest. We run through the causal boosting algorithm again, making all the same splits as in the original training. The difference is in how we estimate the value returned in each node of each shallow causal tree. As in causal forests and in step 3 of the transformed outcome forest, we use $(\mathbf{X}^{(tr)}, \mathbf{T}^{(tr)}, \mathbf{Y}^{(tr)})$ to populate the nodes of the constituent causal trees and estimate the ATE within each node. The residuals r_i from the causal boosting algorithm are initialized to be the y_i from the validation set and are updated according to these re-fitted trees. The result is a new "honest" sequence of models $H_1(x, T), ..., H_K(x, T)$.

We are ready to define our validation error for each of the original models $G_1(x, T), ..., G_K(x, T)$.

The validation error for a causal boosting model with k trees is given by

$$\sum_{x \in v} \left(\{G_k(x,1) - G_k(x,0)\} - \{H_K(x,1) - H_K(x,0)\} \right)^2 .$$

We have several remarks to make about this form. $G_k(x,1) - G_k(x,0)$ is the estimated treatment effect at x, for causal boosting with k trees. $H_K(x,1) - H_K(x,0)$ is the estimated treatment effect correspond to the maximum number of trees, *using the responses from the validation set*. For a large number of trees, we can be sure that this is over-fitting to the response, and this is the analog of traditional cross-validation, which compares predictions on the validation set with observed response in the validation set. This observed response, corresponding to the saturated model, is as over-fitted as possible. Intuitively, we are comparing our estimated treatment effect for each validation point against another estimate, which uses the same structure as the model fit to find similar patients and estimate the treatment effect based on those similar patients, some of whom will have received treatment, some of who will have received control. The better the structure is that causal boosting has learned for the heterogeneous treatment effect, the more the local ATE in the training set will mirror the local ATE in the validation set. For the results in Section 3.7, we use this procedure to do cross-validation for causal boosting.

3.5.2 Within-leaf propensity adjustment

When the goal is to estimate not an ATE but rather an individualized treatment effect, the propensity score methods described in Section 3.2.1 and in the appendix do not immediately extend. Consider for example propensity score stratification. Because each patient belongs to only one stratum of propensity score, we can not average treatment effect estimates for a patient across strata. Technically, if we were to fit a causal boosting model within each stratum, each of these models would be able to make a prediction for the query patient. But then all but one of these models would be unwisely extrapolating outside of its training set to make this prediction. An alternative to propensity score stratification, inverse probability weighting is still viable, but the volatility of this method is exacerbated by the attempt to estimate a varying treatment effect, rather than a constant one.

Within each leaf of a causal tree, however, we estimate an ATE. This is where causal boosting adjusts for non-random treatment assignment, using propensity score stratification to reduce the

bias in the estimate of the within-leaf ATE. Before initiating the causal boosting algorithm, we begin by evaluating the propensity score for each patient, which is an estimate of probability of being assigned the treatment, conditioned on the observed covariates. Any binomial regression technique could be used here. We fit a probability forest [Malley et al., 2012], which is similar to a random forest for classification [Breiman, 2001] except that each tree returns a probability estimate rather than a classification. The trees are combined by averaging the probability estimates and not by majority vote. We denote the treatment assignment probability as a function of the covariates by $\pi(x) \equiv \mathbb{P}(T = 1|X = x)$ and the corresponding propensity scores by $\hat{\pi}_i \equiv \hat{\pi}(x_i)$.

We group the patients into S strata of similar propensity scores denoted $1, ..., S$. For example, there could be $S = 10$ strata, with the first comprising $\hat{\pi} \in [0, 0.1)$ and the last comprising $\hat{\pi} \in [0.9, 1]$, with equal-length intervals in between. We use $s_i \in \{1, ..., S\}$ to denote the stratum to which patient i belongs. Hence the data that we observe within each leaf of a causal tree are of the form $(X_i, s_i, T_i, Y_i) \in \mathbb{R}^p \times \{1, ..., S\} \times \{0, 1\} \times \mathbb{R}$. We use n_ℓ to denote the number of patients in leaf ℓ and index these patients by $i = 1, ..., n_\ell$. The propensity-adjusted ATE estimate in leaf ℓ is given by

$$\hat{\tau}_\ell = \frac{\sum_{s=1}^{S} n_{s\ell}(\bar{Y}_{1s\ell} - \bar{Y}_{0s\ell})}{\sum_{s=1}^{S} n_{s\ell}}, \text{ where } \bar{Y}_{ts\ell} = \frac{\sum_{i=1}^{n_\ell} \mathbb{I}_{\{T_i=t \wedge s_i=s\}} Y_i}{n_{ts\ell}} \tag{3.3}$$

is the mean response among the treatment ($t = 1$) or control ($t = 0$) group in stratum s, and $n_{ts\ell} = \sum_{i=1}^{n_\ell} \mathbb{I}_{\{s_i=s\}}$ is the corresponding number of patients in leaf ℓ for $t \in \{0, 1\}, s \in \{1, ..., S\}$. Finally, $n_{s\ell} = n_{1s\ell} + n_{0s\ell}$.

The estimated variance of $\hat{\tau}_\ell$ is

$$\widehat{\text{Var}}(\hat{\tau}_\ell) = \frac{\sum_{s=1}^{S} n_{s\ell}^2 \hat{\sigma}_{s\ell}^2}{(\sum_{s=1}^{S} n_{s\ell})^2}, \text{ where } \hat{\sigma}_{s\ell}^2 = \frac{s_{1s\ell}^2}{n_{1s\ell}} + \frac{s_{0s\ell}^2}{n_{0s\ell}},$$

and $s_{ts\ell}^2$ is the sample variance of the response for arm t of stratum s in leaf ℓ.

Hence, for two candidate daughter leaves ℓ and r of the same parent, The natural extension of the squared T-statistic splitting criterion from Athey and Imbens [2016] is

$$\frac{|\hat{\tau}_\ell - \hat{\tau}_r|}{\sqrt{\widehat{\text{Var}}(\hat{\tau}_\ell) + \widehat{\text{Var}}(\hat{\tau}_r)}}.$$

This is the propensity-stratified splitting criterion used by causal boosting. This criterion could

also be used by a causal forest as it applies directly to its constituent causal trees.

We use this propensity adjustment not only for determining the split in a causal tree but also for estimating the treatment effect in the node. Specifically, the causal tree returns two values in each leaf: the propensity-adjusted mean response in the treatment and control groups.

$$\frac{\sum_{s=1}^{S} n_{s\ell}\bar{Y}_{1sl}}{\sum_{s=1}^{S} n_{s\ell}} \quad \text{and} \quad \frac{\sum_{s=1}^{S} n_{s\ell}\bar{Y}_{0sl}}{\sum_{s=1}^{S} n_{s\ell}}.$$

3.6 Causal MARS

One drawback to tree-based methods is that because they use the average treatment effect within each leaf as the prediction for that leaf, there could be high bias in this estimate. This is especially problematic when it comes to confidence interval construction for personalized treatment effects. The variance of the estimated treatment effect is relatively straightforward to estimate, but the bias presents more of a challenge.

Multivariate adaptive regression splines (MARS, Friedman [1991]) can be thought of as a modification to CART which alleviates this bias problem. MARS starts with the constant function $f(x) = \beta_0$ and considers adding pairs of functions of the form $\{(x_j - c)_+, (c - x_j)_+\}$ and also the products of variables in the model with these pairs, choosing the pair which lead to the greatest drop in training error when they are added to their model, with regression coefficients estimated via OLS. The difference between this and CART is that in CART the pairs of functions considered are of the form $\{\mathbb{I}_{\{x_j - c \geq 0\}}, \mathbb{I}_{\{c - x_j > 0\}}\}$, and when a product with one of the included terms in chosen, it replaces the included term in the model [Hastie et al., 2009].

We propose causal MARS as the adaptation of MARS to the task of treatment effect estimation. We fit two MARS models in parallel in the two arms (treatment and control) of the data, at each step choosing the same basis functions to add to each model. The criterion that we use identifies the best basis in terms of explaining treatment effect: we compare the drop in training error from including the basis in both models with different coefficients to the drop in training error from including the basis in both models with the *same* coefficient in each model. The steps of causal MARS are as follows. The parameter D controls the maximum dimension of the regression basis, and in practice we use 11 in our examples. Algorithm 3 has the details.

To reduce the variance of causal MARS, we perform bagging by taking B bootstrap samples

Algorithm 3: Causal MARS

1. Define $\mathcal{F} = \{\{(x_j - c)_+, (c - x_j)_+\} : c \in \{\mathbf{X}_{ij}\}, j \in \{1, ..., p\}\}$.
2. Initialize $\mathcal{B} = \{1\}$.
3. For d in $1, ..., D$: (growing the model)
   (a) For each pair of functions $\{f, g\} \in \{\{b(x)f^*(x), b(x)g^*(x)\} : b \in \mathcal{B}, \{f^*, g^*\} \in \mathcal{F}\}$:
      i.
$$RSS_\mu = \min_{\beta^1, \beta^0} \sum_{i=1}^{n} \left(y_i - \sum_{b \in \mathcal{B}} (\beta_b^1 b(x_i) \mathbb{I}_{\{t_i=1\}} + \beta_b^0 b(x_i) \mathbb{I}_{\{t_i=0\}}) - \sum_{h \in \{f,g\}} \beta_h h(x_i) \right)^2$$
      ii.
$$RSS_\tau = \min_{\beta^1, \beta^0} \sum_{i=1}^{n} \left(y_i - \sum_{b \in \mathcal{B}} (\beta_b^1 b(x_i) \mathbb{I}_{\{t_i=1\}} + \beta_b^0 b(x_i) \mathbb{I}_{\{t_i=0\}}) - \sum_{h \in \{f,g\}} (\beta_h^1 h(x_i) \mathbb{I}_{\{t_i=1\}} + \beta_h^0 h(x_i) \mathbb{I}_{\{t_i=0\}}) \right)^2$$
      iii.
$$dRSS = RSS_\tau - RSS_\mu$$
   (b) Choose $\{f, g\}$ which maximize $dRSS$ and add them to $\mathcal{B}$.
4. Backward deletion: delete terms one at a time, using the same criterion as in the forward stepwise 3(a). Use the out-of-bag error to estimate the optimal model size.

of the original dataset and fitting the causal MARS model to each one. The estimated treatment effect for an individual is the average of the estimates for this individual by the B models.

Note that the algorithm described above applies to the randomized case, not observational data. Given S propensity strata and membership $s \in 1, ..., S$, for each patient, we use the same basis functions within each stratum but different regression coefficients. Within each stratum, the coefficients are estimated separately from the coefficients in other strata. Given the entry criterion $dRSS_s$ and number of patients n_s in each stratum, we combine these into a single criterion $\sum_s n_s dRSS_s$. This is the *propensity-adjusted* causal MARS.

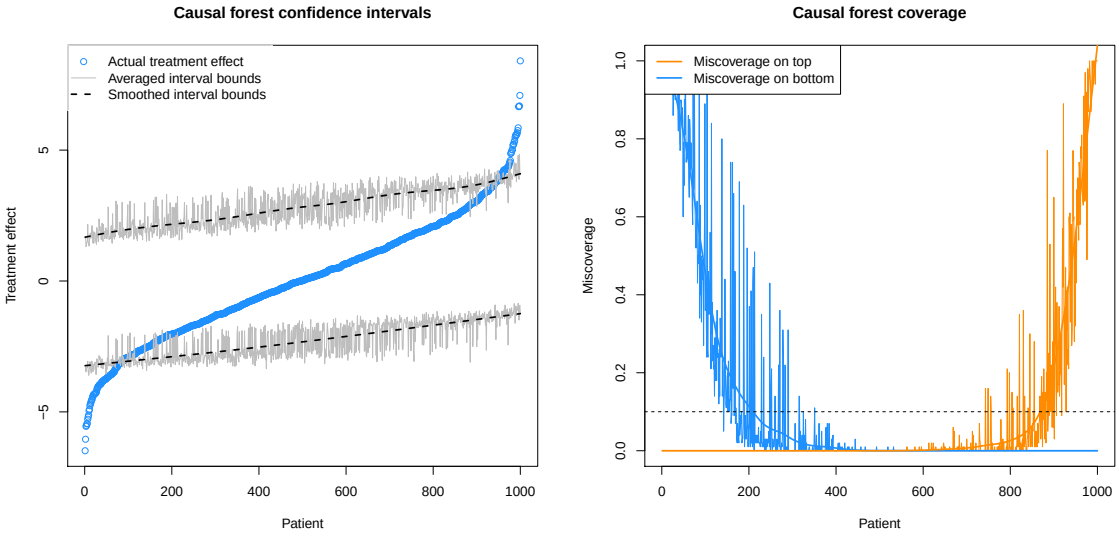
3.6.1 Confidence intervals

One advantage of the bagging-based methods—causal forest and causal MARS—is that in the process of computing the treatment-effect estimates, one gets at no extra cost the computations necessary to estimate the variance of the estimators. Each of the bagged models is based on its own bootstrap re-sampling of the data, so for each patient we have B re-sampled treatment effect estimates, where B is the number of bags. We propose using the quantiles of these estimates as the confidence interval for each patient. To construct a $(1 - \alpha)$ confidence interval, we use the $\alpha/2$ and $1 - \alpha/2$ quantiles of the bootstrapped estimates as lower and upper bounds, respectively.

Note that this procedure is targeted at the variability of a single causal tree or a single causal MARS model, but the methods we propose involve averaging these models to reduce their variance. This will make our intervals more conservative because the variance of the bagged models will be lower than the variance of the individual models. However, as the results in this section demonstrate, the conservative nature of these confidence intervals helps with coverage problems due to the inability to fully remove the bias from the treatment effect estimates.

Figure 3.3 shows confidence interval results for causal forest applied to Simulation 8 in Section 3.7. That section describes in detail our simulation scheme, but in this section we use it only as an illustration of the confidence interval results. The left figure shows the average upper and lower bounds of the confidence interval for each patient, across 100 simulations. This demonstrates the difficulty with constructing confidence intervals for random forest predictions: Because of the relatively high bias from using the average as the estimate within each leaf, the confidence intervals do not come close to maintaining $(1 - \alpha)$ coverage for patients with relatively small or relatively large treatment effects.

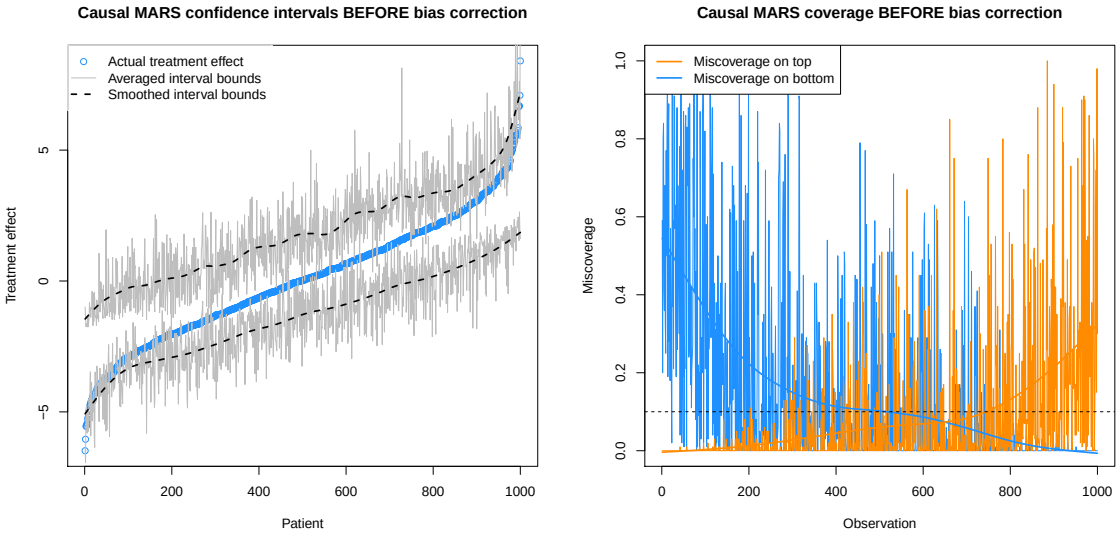
Figure 3.3: Confidence intervals for causal forest in Scenario 8 from Section 3.7. On the left in blue we plot the true treatment effect for each patient against the index of the patient, sorted by treatment effect. The thin gray lines show the average upper and lower confidence interval bounds for each patient, and the dotted black line smooths over these averages. On the left the thin lines give the miscoverage rate for each patient, and the thick lines smooth over these thin lines. These results reflect 100 simulations using 50 bagged causal trees.



This problem for causal forests was the motivation for the development of causal MARS. By using piecewise linear models instead of piecewise constant models, MARS can achieve lower bias than regression trees, which is important for bootstrap confidence-interval construction. Figure 3.4 shows the results of constructing confidence intervals for the causal MARS estimates in a single simulation. The average confidence intervals are more volatile in Figure 3.4 than in Figure 3.3 because causal MARS is a higher-variance method. But we see that the confidence intervals adhere more closely to the true treatment effect for this method than for the causal forest. Examining the coverage, we see that there is still a bias problem for treatment effects near the edges of the range of values, but the miscoverage is closer to 0.5, an improvement of the coverage which approaches 1 for causal forest.

Still, bagged causal MARS has not fully mitigated the bias problem. We see that the miscoverage on bottom is decreasing with the true treatment effect, and the miscoverage on top is increasing with the true treatment effect. We attempted to address this with a bias correction. We bootstrapped residuals from the fitted model and applied a standard bootstrap bias correction. The results of

Figure 3.4: Confidence intervals for causal MARS in Scenario 8 from Section 3.7. On the left in blue we plot the true treatment effect for each patient against the index of the patient, sorted by treatment effect. The thin gray lines show the average upper and lower confidence interval bounds for each patient across 100 simulations, and the dotted black line smooths over these averages. On the left the thin lines give the miscoverage rate for each patient, and the thick lines smooth over these thin lines. These results reflect 100 simulations using 50 bagged causal MARS models.



this correction are shown in Figure 3.5. Here the confidence intervals adhere even more closely to the true treatment effect, and the coverage is improved. The miscoverage on either side of the confidence interval is capped at 0.2 when smoothed, though the target miscoverage rate is 0.1. We have taken steps toward constructing confidence intervals for personalized treatment effects, but it remains an area for future research.

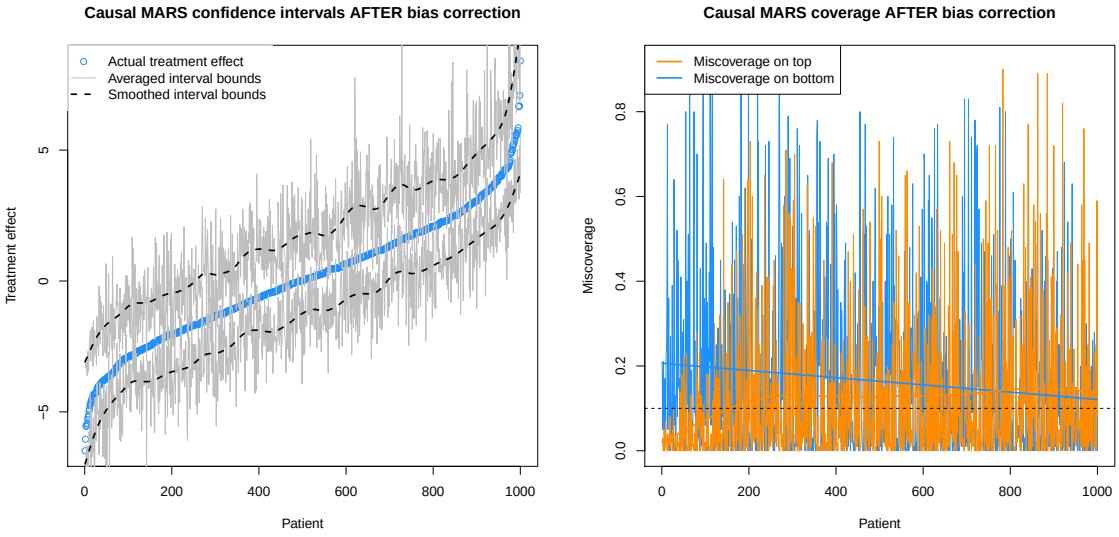
3.7 Simulation study

In the design of our simulations to evaluate performance of methods for heterogeneous treatment effect estimation, there are four elements to the generation of synthetic data:

1. The number n of patients in the training set, and the number p of features observed for each patient.

2. The distribution $\mathcal{D}_X$ of the feature vectors X_i. Across all scenarios, we draw odd-numbered

Figure 3.5: Bias-corrected confidence intervals for causal MARS in Scenario 8 from Section 3.7. On the left in blue we plot the true treatment effect for each patient against the index of the patient, sorted by treatment effect. The thin gray lines show the average upper and lower confidence interval bounds for each patient across 100 simulations, and the dotted black line smooths over these averages. On the left the thin lines give the miscoverage rate for each patient, and the thick lines smooth over these thin lines. These results reflect 100 simulations using 50 bagged causal MARS models.



features independently from a standard Gaussian distribution. We draw even-numbered features independently from a standard Bernoulli distribution.

3. The propensity function $\pi(\cdot)$, the mean effect function $\mu(\cdot)$ and the treatment effect function $\tau(\cdot)$. We take the conditional mean effect functions to be $\mu_1(x) = \mu(x) + \tau(x)/2$ and $\mu_0(x) = \mu(x) - \tau(x)/2$.

4. The conditional variance σ_Y^2 of Y_i given X_i and T_i. This corresponds to the noise level, and we choose is to make the percentage of null variance explained of the true model to be roughly 20-25%. This ensures we are comparing the methods on relevant simulations.

Given the elements above, our data generation model is, for $i = 1, ..., n$:

$$X_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}_X$$

$$T_i \overset{\text{ind.}}{\sim} \text{Bernoulli}(\pi(X_i))$$

$$Y_i \overset{\text{ind.}}{\sim} \text{Normal}\left(\mu(X_i) + (T_i - 1/2)\tau(X_i), \sigma_Y^2\right)$$

The third element above, encompassing $\pi(\cdot)$, $\mu(\cdot)$ and $\tau(\cdot)$, is most interesting. Note that $\pi(\cdot)$ and $\mu(\cdot)$ are nuisance functions, and $\tau(\cdot)$ is the function we are interested in estimating. In this section, we present two batches of simulations, the first of which represent randomized experiments. The second batch of simulations represent observational studies. Within each set of simulations, we make eight different choices of mean effect function and treatment effect function, meant to represent a wide variety of functional forms: both univariate and multivariate; both additive and interactive; both univariate and multivariate. The eight functions that we chose are:

$$f_1(x) = 0 \qquad f_2(x) = 5\mathbb{I}_{\{x_1>1\}} - 5 \qquad f_3(x) = 2x_1 - 4$$

$$f_4(x) = x_2x_4x_6 + 2x_2x_4(1-x_6) + 3x_2(1-x_4)x_6 + 4x_2(1-x_4)(1-x_6) + 5(1-x_2)x_4x_6$$
$$+6(1-x_2)x_4(1-x_6) + 7(1-x_2)(1-x_4)x_6 + 8(1-x_2)(1-x_4)(1-x_6)$$

$$f_5(x) = x_1 + x_3 + x_5 + x_7 + x_8 + x_9 - 2$$

$$f_6(x) = 4\mathbb{I}_{\{x_1>1\}}\mathbb{I}_{\{x_3>0\}} + 4\mathbb{I}_{\{x_5>1\}}\mathbb{I}_{\{x_7>0\}} + 2x_8x_9$$

$$f_7(x) = \frac{1}{2}\left(x_1^2 + x_2 + x_3^2 + x_4 + x_5^2 + x_6 + x_7^2 + x_8 + x_9^2 - 11\right)$$

$$f_8(x) = \frac{1}{\sqrt{2}}\left(f_4(x) + f_5(x)\right)$$

Each of the eight functions above is centered and scaled so that with respect to the distribution $\mathcal{D}_X$, each has mean close to zero and all have roughly the same variance. Table 3.1 gives the mean and treatment effect functions for the eight randomized simulations, in terms of the eight functions above. In these simulations $\pi(x) = 1/2$ for all $x \in \mathbb{R}^p$. In addition to the methods described in Sections 3.4, 3.5 and 3.6, we include results for two additional estimators for comparison. The null estimator is simply the difference $\bar{Y}_1 - \bar{Y}_0$ in mean response between treated and untreated patients.

Table 3.1: Specifications for the 16 simulation scenarios. The four rows of the table correspond, respectively, to the sample size, dimensionality, mean effect function, treatment effect function and noise level. Simulations 1 through 8 use randomized treatment assignment, meaning $\pi(x) = 1/2$. Simulations 9 through 16 have a bias in treatment assignment, specified by (3.4).

Scenarios	1, 9	2, 10	3, 11	4, 12	5, 13	6, 14	7, 15	8, 16
n	200	200	300	300	400	400	1000	1000
p	400	400	300	300	200	200	100	100
$\mu(x)$	$f_8(x)$	$f_5(x)$	$f_4(x)$	$f_7(x)$	$f_3(x)$	$f_1(x)$	$f_2(x)$	$f_6(x)$
$\tau(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$
σ_Y^2	1	1/4	1	1/4	1	1	4	4

This provides a naive baseline. The other competitor is the gradient forest of Athey et al. [2017], using the `gradient.forest` R package made available online by the authors. The results of the fist batch of simulations are shown in Figure 3.6.

If we pick "winners" in each of the simulation scenario based on which method has the lowest distribution of errors, causal MARS would win Scenarios 5, 7 and 8, tying with the pollinated transformed outcome forest in Scenario 4. The PTO forest would win Scenarios 2 and 3, tying with causal boosting in Scenario 6. In general all of the methods outperform the null estimator except in Scenario 1, when the treatment effect is constant, and in Scenario 6, when the gradient forest perform worst.

The second batch of simulations matches the parameters listed in Table 3.1: Scenario 9 is like Scenario 1; Scenario 10 is like Scenario 2; and so on. The difference is in the propensity function. For this second batch of simulations, we use

$$\pi(x) = \frac{e^{\mu(x)-\tau(x)/2}}{1 + e^{\mu(x)-\tau(x)/2}}. \tag{3.4}$$

The interpretation of this propensity function is that patients with greater mean effect are more likely to receive the treatment. This resembles a situation in which greater values of the outcome are worse for the patient, and only patients who have need for treatment will receive it. There are many possible forms for the propensity function, but we focus on this one because it is particularly troublesome, and a good estimator of the treatment effect needs to avoid the pitfall of estimating to great an effect because the treated patients have greater mean effect. This is exactly the kind of bias we are most concerned about in observational studies. The results of this second batch of simulations are shown in Figure 3.7.

Figure 3.6: Results across eight simulated randomized experiments. For details of the generating distributions, see Table 3.1. The seven estimators being evaluated are: NULL = the null prediction, GF = gradient forest, PTO0 = pollinated transformed outcome forest (using propensity = 1/2), CB0 = causal boosting, CM0 = causal MARS. The vertical blue bar shows the standard deviation of the response, for assessing the practical significance of the difference between the methods' performances.

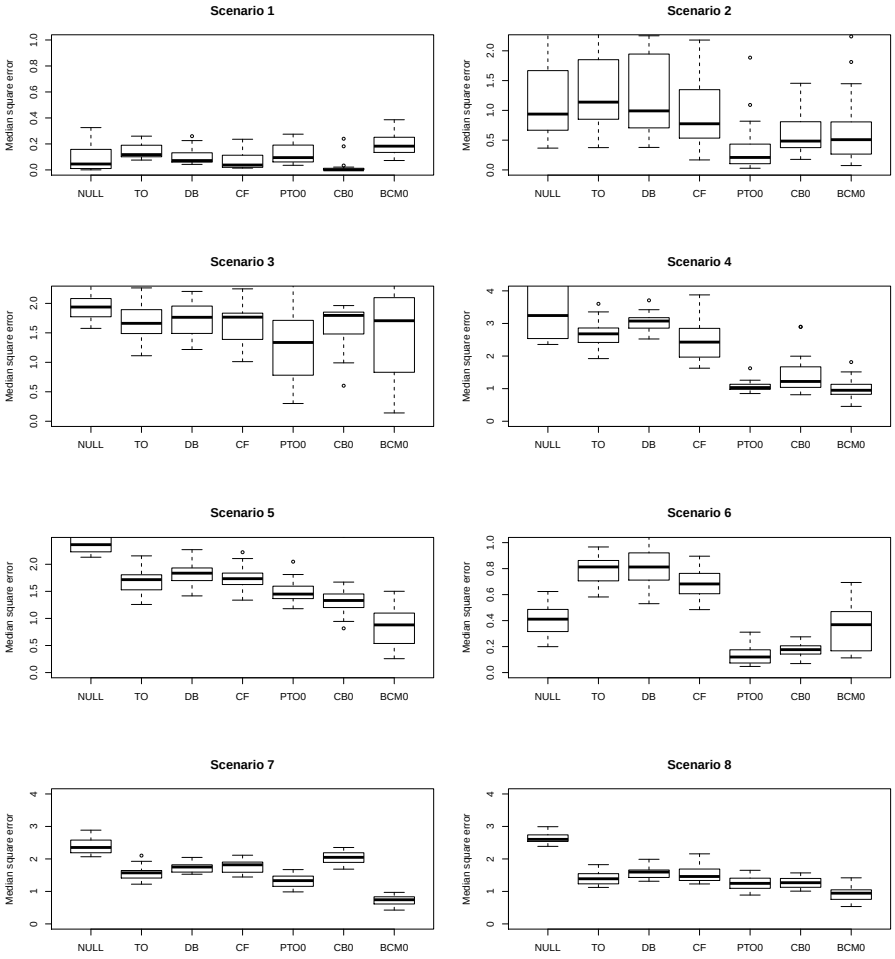
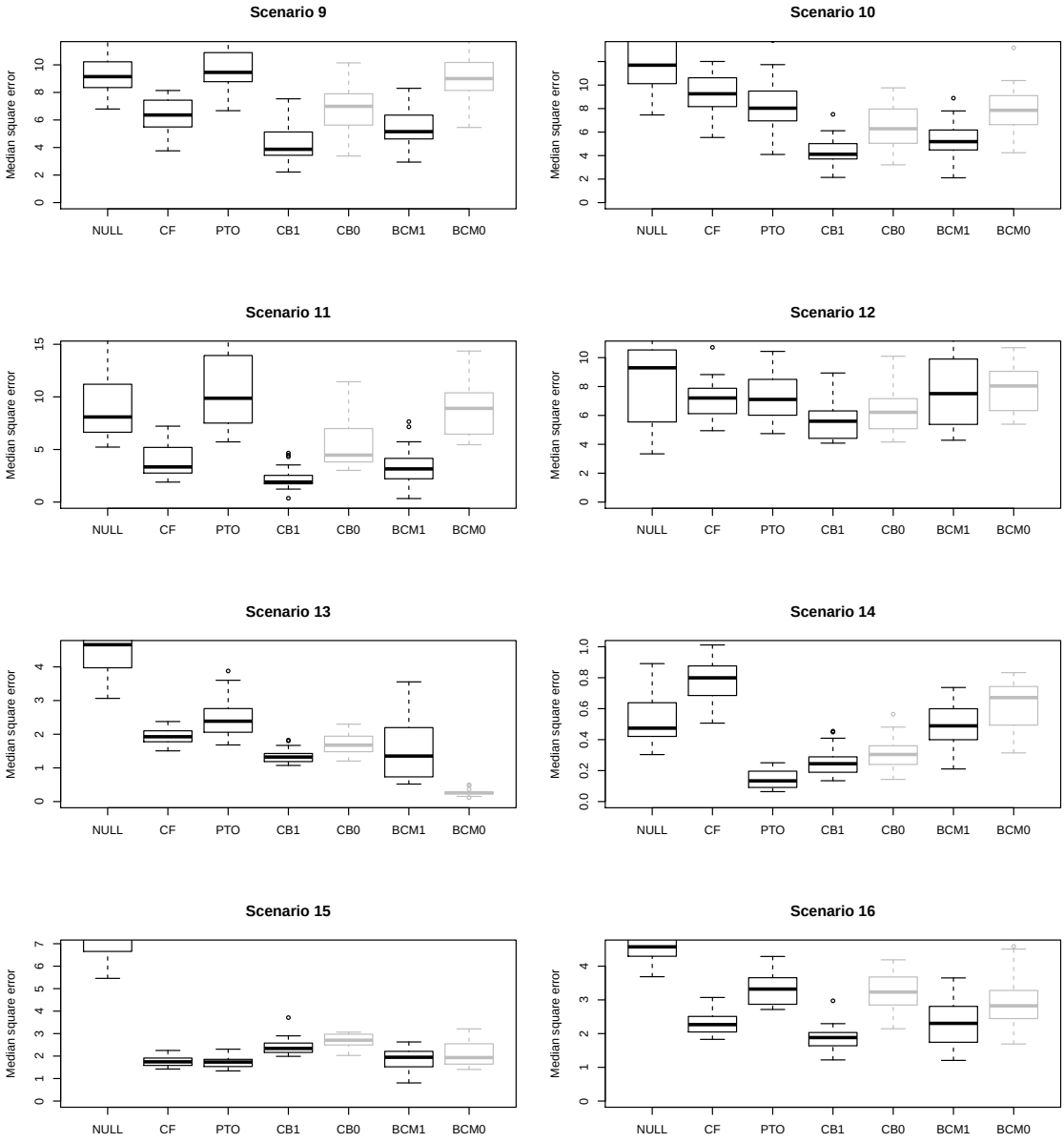


Figure 3.7: Results across eight simulated observational studies, in which treatment is more likely to be assigned to those with a greater mean effect. The seven estimators being evaluated are: NULL = the null prediction, GF = gradient forest, PTO = pollinated transformed outcome forest, CB1 = causal boosting (propensity adjusted), CB0 = causal boosting, CM1 = causal MARS (propensity adjusted), CM0 = causal MARS. CB0 and CM0 are in gray because they would not be used in this setting. They are provided for reference to assess the effect of the propensity adjustment. The vertical blue bar shows the standard deviation of the response, for assessing the practical significance of the difference between the methods' performances.



In the batch of simulations with biased treatment assignments, propensity-adjusted causal boosting shines. In six of the eight simulations, causal boosting as either the lowest error distribution or is one of the two methods with the lowest error distribution. Curiously, in Scenario 13, unadjusted causal MARS performs very well, but the propensity adjustment ruins this performance. In Scenario 15, PTO forest and gradient forest produce the best results though all of the methods perform well. Overall, across the 16 simulation scenarios, causal boosting and causal MARS stand out as having the best performance.

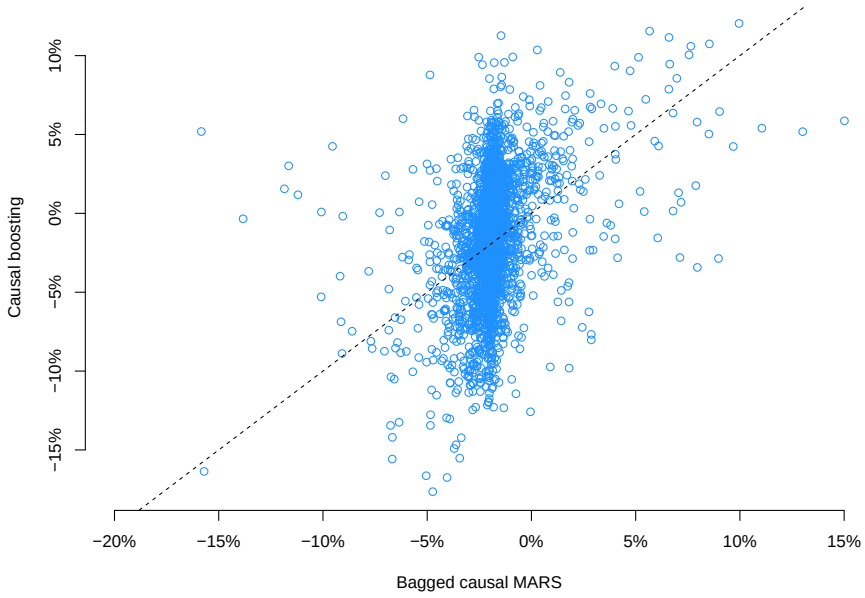
3.8 Application

In September 2016, *New England Journal of Medicine* opened The SPRINT Data Analysis Challenge, based on the complete dataset from a randomized trial of a novel intervention for the treatment of high blood pressure [SPRINT Research Group, 2015]. The goal was open-ended: to draw novel or clinically useful insights from the SPRINT dataset, possibly in tandem with other publicly available data.

The intervention in the randomized trial [SPRINT Research Group, 2015] was a more intensive control of systolic blood pressure (target 120 mm Hg) than is standard (target 140 mm Hg). The primary outcome of interest was whether the patient experienced any of the following events: myocardial infarction (heart attack), other acute coronary syndrome, stroke, heart failure or death from cardiovascular causes. The trial, which enrolled 9361 patients, ended after a median follow-up period of 3.26 years, when researchers determined at a pre-planned checkpoint that the population-average outcome for the intensive treatment group (1.65% incidence per year) was significantly better than that of the standard treatment group (2.19% incidence per year).

In addition to the primary event, for each patient researchers tracked several other adverse events, as well as 20 baseline covariates recorded at the moment of treatment assignment randomization: 3 demographic variables, 6 medical history variables and 11 lab measurements. The question that we seek to answer in this section is whether we can use these variables to give personalized estimates of treatment effect which are more informative than the population-level average treatment effect. To answer this question, we apply the gradient forest and causal MARS to these data.

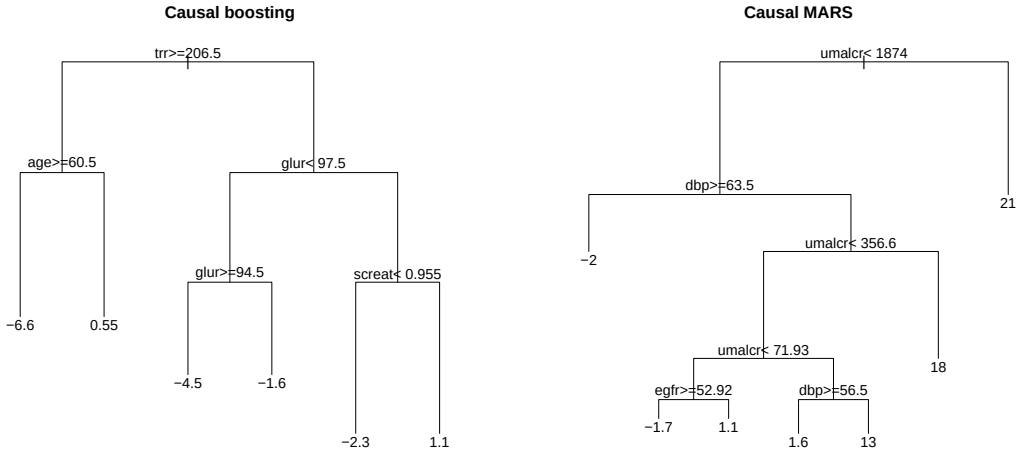
Figure 3.8: Personalized treatment effect estimates from causal boosting and causal MARS. Each circle represents a patient, who gets a personalized estimate from each method. The dashed line represents the diagonal, along which the two estimates are the same.



Of the 9361 patients who underwent randomization, 1172 (12.5%) died, discontinued intervention, withdrew consent or were lost to follow-up before the conclusion of the trial. There is little evidence (χ^2 p-value = 31%) that this censorship was more common in either arm of the trial. To extract a binary outcome from these survival data, we use as our response the indicator that a patient experiences the primary outcome within 1000 days of beginning treatment, ignoring patients who were censored before 1000 days. Additionally, we dropped the 1.8% of patients who have at least one lab measure missing. This leaves us with a sample of 7344 patients, which we split into equally sized training and validation sets.

The results of fitting causal boosting and causal MARS on the training sample of 3672 patients are shown in Figure 3.8. We observe that the two methods yield very different distributions of estimated personalized treatment effects in the aggregate. Causal boosting produces estimates resembling a normal distribution with a standard deviation of about 3.5% risk. In contrast, causal MARS estimates almost all patients to have a treatment effect between −5% risk and +0% risk, but for a small percentage of patients the treatment effect is much greater or much lesser. The tails of this distribution are much heavier than that of a normal distribution. In fact, a very small number

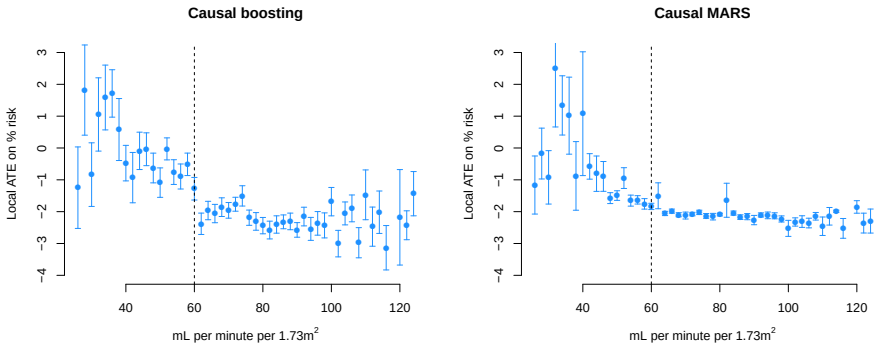
Figure 3.9: Decision trees summarizing with broad strokes the inferences of causal boosting and causal MARS. The variables are: `trr` triglcerides (mg/dL) from blood draw; `age` (years) age at beginning of trial; `glur` glucose (mg/dL) from blood draw; `screat` creatinine (mg/dL) from blood draw; `umalcr` albumin/creatinine ratio from urine sample; `dbp` diastolic blood pressure (mm Hg); `egfr` estimated glomerular filtration rate (mL/min/1.73m^2). If the inequality at a split is true for a patient, then that patient belongs to the left daughter node.



of patients (0.4% of the training sample) are not included in this figure because their treatment effect estimate from causal MARS falls outside of the plotted region.

Figure 3.9 depicts decision trees which summarize the key inferences made by causal boosting and causal MARS. Each leaf gives the average estimated treatment effect for patients who belong to that leaf. Such a decision could be reported to a physician to explain the basis for these personalized treatment effect estimates. According to causal boosting, for example, older patients with high triglycerides stand to gain more from the intensive blood pressure treatment than younger patients with high triglycerides. Among patients with low triglycerides and high glucose, those with low creatinine stand to benefit more from the intensive treatment than those with high creatinine. The decision tree for causal MARS makes the extreme claim that for patients with urine albumin/creatinine ratio above 1874, the average treatment effect is a 21% increase in risk. Discussions with practitioners suggest that the distribution of personalized treatment effects estimated by causal boosting is more plausible than that of causal MARS. As such, we focus our interpretation on the results of causal boosting for the reminder of this section.

Figure 3.10: Training set personalized treatment effects, estimated via causal boosting, versus estimated glomerular filtration rate (eGFR). Patients are stratified according to eGFR on the x-axis, and each point gives the average personalized treatment effect among patients in that stratum. Error bars correspond to one standard error for the mean personalized treatment effect. The vertical dashed line represents a medical cutoff, below which patients are considered to suffer from chronic kidney disease.

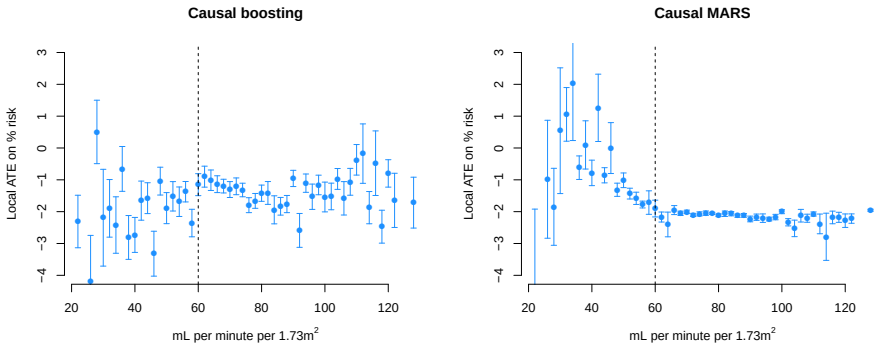


To simplify the results even more than the decision tree does, we note that for both causal boosting and causal MARS, the two features which correlate most to the personalized treatment effect estimates are estimated glomerular filtration rate (eGFR) and creatinine. These two variables are highly correlated with each other, as creatinine is one of the variables used to estimate GFR. Both are used to assess kidney health, and patients with eGFR below 60 are considered to have chronic kidney disease. Figure 3.10 shows the relationship between eGFR and the estimated personalized treatment effect from causal boosting. Despite there being no manual notation in the data that there is something special about an eGFR of 60, we have learned from causal boosting that patients below this cutoff have less to gain from the intensive blood pressure treatment than patients above this cutoff.

Note that we are not only interested in whether a patient's personalized treatment effect is positive or negative. Intensive control of blood pressure comes with side effects and should only be assigned to patients for whom the benefit of reducing the risk of an adverse coronary event is substantial. The results of causal boosting on the training set would suggest that patients with chronic kidney disease have less to gain from this treatment than do other patients.

The results above tell an interesting story: If you are a patient with chronic kidney disease (eGFR < 60), you are expected to benefit less from intensive blood pressure control. As discussed in Section 3.5.1, validating treatment effect estimates is challenging because we do not observe the

Figure 3.11: Validation set personalized treatment effects, estimated via causal boosting, versus estimated glomerular filtration rate (eGFR). Patients are stratified according to eGFR on the x-axis, and each point gives the average personalized treatment effect among patients in that stratum. Error bars correspond to one standard error for the mean personalized treatment effect. The vertical dashed line represents a medical cutoff, below which patients are considered to suffer from chronic kidney disease.



treatment effect for any individual patient. In this section, we make an attempt to validate the more general conclusion from the previous section: that the treatment has less benefit for patients with chronic kidney disease.

Figure 3.11 shows the results of fitting causal boosting on the held-out validation set of 3672 patients. We see that the relationship between eGFR and estimated treatment effect does not tell the same story as in the training set. In fact, there is no clear relationship between these two variables in the validation set.

It is possible that we have insufficient power in the validation set to identify the relationship between eGFR and treatment effect and that with a larger sample of patients, we would have validated our conclusions from the training set. It is worth noting that the team from Boston University which placed second in the SPRINT Data Analysis Challenge made the same finding as shown in the causal boosting results. They found that intensive blood pressure management does not improve primary outcomes for patients with chronic kidney disease. Something that the authors do not address is why they chose to analyze patients with chronic kidney disease. Presumably they used some combination of prior medical knowledge and manual hypothesis selection. In our training set, we came to the same conclusion using causal boosting without the benefit of either of these steps. The dissimilar results on the validation set could be explained by insufficient power.

3.9 Discussion

We have proposed and compared a number of different methods for estimating heterogeneous treatment effects from high-dimensional covariates. The causal boosting and causal MARS approaches seem particularly promising. More work is needed in refining and testing these methods, and in the construction of reliable confidence intervals for the estimated effects.

3.10 Appendix

In this appendix we outline the already-established techniques for using propensity score to adjust for bias in treatment assignment for observational studies in which the goal is to estimate a population-average treatment effect (ATE). Define $f(x)$ the marginal feature density, $f_1(x)$ the conditional density of X given $T = 1$ (and likewise $f_0(x)$), where T is binary treatment indicator, and let $\pi_1 = \mathbb{P}(T = 1)$ be the marginal proportion of treated. Let $\mu_1(X) = \mathbb{E}[Y|T = 1, X]$, and likewise $\mu_0(X)$, and $\tau(X) = \mu_1(X) - \mu_0(X)$. Finally, let $\pi(X) = \mathbb{P}(T = 1|X)$ be the treatment propensity.

Transformed outcome averaging

Note that the *transformed outcome*

$$Z \equiv T\frac{Y}{\pi(X)} + (1 - T)\frac{-Y}{1 - \pi(X)}$$

satisfies

$$\begin{aligned}\mathbb{E}[Z|X] &= \mathbb{P}(T = 1|X)\frac{1}{\pi(x)}\mathbb{E}[Y|T = 1, X] - \mathbb{P}(T = 0|X)\frac{1}{1 - \pi(x)}\mathbb{E}[Y|T = 1, X] \\ &= \mathbb{E}[Y|T = 1, X] - \mathbb{E}[Y|T = 0, X] = \mu_1(X) - \mu_0(X) = \tau(X).\end{aligned}$$

Hence if the expectation of Z is evaluated with respect to the distribution of X,

$$\mathbb{E}_X[Z] = \mathbb{E}_X[\mathbb{E}[Z|X]] = \mathbb{E}_X[\tau(X)].$$

In other words, the transformed outcome is unbiased for the ATE. So a natural estimator for the

ATE in a sample of patients would be the sample mean of the transformed outcome. This justifies for example using Z as a response to grow a random forest in our pollinated transformed outcome forest.

Propensity score stratification

Note that it is not necessarily the case that $E[Y|T = 1] = E[\mu_1(X)|T = 1]$ and $E_X[\mu_1(X)]$ are the same; it is possible that conditioning on T changes the distribution of X and consequently the distribution of $\mu_1(X)$. This is the essence of why we cannot ignore non-randomized treatment assignment in observational studies. However, it is the case that

$$\mathbb{E}[Y|T = 1, \pi(X)] = \mathbb{E}[\mu_1(X)|\pi(X)].$$

To see this, note that $X \perp\!\!\!\perp T|\pi(X)$ because by assumption $T \sim \text{Binomial}(1, \pi(X))$. Hence the conditional distribution of X given $\pi(X)$ and T is the same as the conditional distribution of X given $\pi(X)$. This implies that

$$\mathbb{E}[Y|T = 1, \pi(X)] = \mathbb{E}\left\{\mathbb{E}[Y|T = 1, X] \;\middle|\; T = 1, \pi(X)\right\} = \mathbb{E}[\mu_1(X)|\pi(X)].$$

What this says is that for fixed $\pi(X)$, the mean response under treatment is unbiased for the conditional expectation of $\mu_1(X)$. This equality holds for any value of X, so the expectations of these two quantities are the same with respect to the distribution of $\pi(X)$:

$$\mathbb{E}_{\pi(X)}\left[\mathbb{E}[Y|T = 1, \pi(X)]\right] = \mathbb{E}_{\pi(X)}\left[\mathbb{E}[\mu_1(X)|\pi(X)]\right] = \mathbb{E}_X[\mu_1(X)].$$

This leads to the following estimator for $\mathbb{E}_X[\mu_1(X)]$: Compute the average response for all treated patients for each value of the propensity, and integrate with respect to the distribution of the propensity. In practice, we approximate this by using a rough approximation to the distribution of $\pi(X)$: Define strata (or bins) of the propensity score, for example (0, 0.1], ..., (0.9, 1). Within each stratum, find the average response among treated patients. Then combine these values in a weighted average, weighting according to the frequency of each stratum. This is our estimate of $\mathbb{E}_X[\mu_1(X)]$. We follow the same procedure in the control arm to estimate $\mathbb{E}_X[\mu_0(X)]$, and the difference is our estimate of $\mathbb{E}_X[\tau(X)]$.

Inverse probability weighting

From Bayes' theorem, $f_1(x) = f(x)\pi(x)/\pi_1$. Consider weighting this density with weights proportional to $1/\pi(x)$. The density of this weighted distribution is given by

$$\tilde{f}_1(x) = \frac{\frac{1}{\pi(x)} f(x)\pi(x)/\pi_1}{\int_{\mathbb{R}} \frac{1}{\pi(x)} f(x)\pi(x)/\pi_1} dx = \frac{f(x)/\pi_1}{\int_{\mathbb{R}} f(x)/\pi_1 dx} = \frac{f(x)/\pi_1}{1/\pi_1} = f(x).$$

Hence the weighted conditional distribution of X given $T = 1$ is the same as the marginal distribution of X. So the expectation of any function of X with respect to this distribution is the same as with respect to the marginal distribution of X. Specifically, using $\tilde{X}$ to denote the random variable following the weighted density $\tilde{f}_1(x)$,

$$\mathbb{E}_{\tilde{X}}[\mu_1(\tilde{X})] = \mathbb{E}_X[\mu_1(X)].$$

Based on this result, we use the sample mean of the response in the treatment arm, with weights proportional to the inverse of the propensity, as an unbiased estimator for $\mathbb{E}_X[\mu_1(X)]$. Similarly, in the control arm we use weights proportional to $1/(1 - \pi(x))$ to get an unbiased estimate for $\mathbb{E}_X[\mu_0(x)]$. The difference between these two is our estimate for $\mathbb{E}_X[\tau(X)]$

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. **TensorFlow**: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, Berkeley, CA, USA, 2016. USENIX Association. ISBN 978-1-931971-33-1. URL `http://dl.acm.org/citation.cfm?id=3026877.3026899`.

Matthew Aguirre, Yosuke Tanigawa, Guhan Venkataraman, Rob Tibshirani, Trevor Hastie, and Manuel A Rivas. Polygenic risk modeling with latent trait-related genetic components. *BioRxiv*, page 808675, 2019.

Theodore Wilbur Anderson et al. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *The Annals of Mathematical Statistics*, 22(3):327–351, 1951.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of the Sciences*, 113(27):7353–7360, 2016.

Susan Athey, Julie Tibshirani, and Stefan Wager. Solving heterogeneous estimating equations with gradient forests. https://arxiv.org/abs/1610.01271, 2017.

Peter C Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research*, 46:399–424, 2011.

Francis R Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9(Jun):1179–1225, 2008.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 08 2009. doi: 10.1214/08-AOS620. URL `https://doi.org/10.1214/08-AOS620`.

Marco Bonetti and Richard D Gelber. Patterns of treatment effects in subsets of patients in clinical trials. *Biostatistics*, 5(3):465–481, 2004.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Daniel P Bovet and Marco Cesati. *Understanding the Linux Kernel: from I/O ports to process management*. " O'Reilly Media, Inc.", 2005.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232–253, 03 2011. doi: 10.1214/10-AOAS388. URL `https://doi.org/10.1214/10-AOAS388`.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T. Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, Adrian Cortes, Samantha Welsh, Alan Young, Mark Effingham, Gil McVean, Stephen Leslie, Naomi Allen, Peter Donnelly, and Jonathan Marchini. The UK biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0579-z. URL `https://doi.org/10.1038/s41586-018-0579-z`.

Christopher C Chang, Carson C Chow, Laurent CAM Tellier, Shashaank Vattikuti, Shaun M Purcell, and James J Lee. Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience*, 4(1), 02 2015. ISSN 2047-217X. doi: 10.1186/s13742-015-0047-8. URL `https://doi.org/10.1186/s13742-015-0047-8`.

Lisha Chen and Jianhua Z Huang. Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, 107(500):1533–1545, 2012.

Wei Chen, Debashis Ghosh, Trivellore E Raghunathan, Maxim Norkin, Daniel J Sargent, and Gerold Bepler. On Bayesian methods of exploring qualitative interactions for targeted treatment. *Statistics in Medicine*, 31(28):3693–3707, 2012.

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian CART model search. *Journal of the American Statistical Assocation*, 93(443):935–948, 1998.

Moody T Chu and Nickolay T Trendafilov. On a differential equation approach to the weighted orthogonal procrustes problem. *Statistics and Computing*, 8(2):125–133, 1998.

Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25, 2010.

Corinna Cortes and Mehryar Mohri. Confidence intervals for the area under the roc curve. In *Advances in Neural Information Processing Systems*, pages 305–312, 2005.

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972. ISSN 00359246. URL `http://www.jstor.org/stable/2985181`.

Richard K Crump, V Joseph Hotz, Guido W Imbens, and Oscar A Mitnik. Nonparametric tests for treatment effect heterogeneity. *The Review of Economics and Statistics*, 90(3):389–405, 2008.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008. ISSN 0001-0782. doi: 10.1145/1327452.1327492. URL `http://doi.acm.org/10.1145/1327452.1327492`.

Christopher DeBoever, Yosuke Tanigawa, Malene E. Lindholm, Greg McInnes, Adam Lavertu, Erik Ingelsson, Chris Chang, Euan A. Ashley, Carlos D. Bustamante, Mark J. Daly, and Manuel A. Rivas. Medical relevance of protein-truncating variants across 337,205 individuals in the uk biobank study. *Nature Communications*, 9(1):1612, 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-03910-9. URL `https://doi.org/10.1038/s41467-018-03910-9`.

Elizabeth R. DeLong, David M. DeLong, and Daniel L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44(3):837–845, 1988. ISSN 0006341X, 15410420. URL `http://www.jstor.org/stable/2531595`.

Benjamin Dubois, Jean-François Delmas, and Guillaume Obozinski. Fast algorithms for sparse reduced-rank regression. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2415–2424. PMLR, 16–18 Apr 2019. URL `http://proceedings.mlr.press/v89/dubois19a.html`.

J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3): 592–606, March 2012. ISSN 0018-9286. doi: 10.1109/TAC.2011.2161027.

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57 (3):592–606, 2011.

Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*, volume 5. Cambridge University Press, 2016.

Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani. Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219*, 2010.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008. doi: 10.1111/j.1467-9868.2008.00674.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2008.00674.x`.

Jerome Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

Jerome Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer series in statistics. Springer-Verlag, 2009. doi: 10.1007/978-0-387-84858-7.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, Articles*, 33(1):1–22, 2010a. ISSN 1548-7660. doi: 10.18637/jss.v033.i01. URL `https://www.jstatsoft.org/v033/i01`.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent, 2010b. ISSN 1548-7660. URL `https://www.jstatsoft.org/v033/i01`.

Karl Ruben Gabriel. The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58(3):453–467, 1971.

M Gail and Richard Simon. Testing for qualitative interactions between treatment effects and patient subsets. *Biometrics*, 41(2):361–372, 1985.

Tian Ge, Chia-Yen Chen, Yang Ni, Yen-Chen Anne Feng, and Jordan W Smoller. Polygenic prediction via bayesian regression and continuous shrinkage priors. *Nature Communications*, 10 (1):1–10, 2019.

John C Gower, Sugnet Gardner Lubbe, and Niel J Le Roux. *Understanding biplots*. John Wiley & Sons, 2011.

Donald P Green and Holger L Kern. Modeling heterogeneous treatment effects in survey experiments with Bayesian additive regression trees. *Public Opinion Quarterly*, 76(3):491–511, 2012.

Eitan Greenshtein and Ya'Acov Ritov. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10(6):971–988, 12 2004. doi: 10.3150/bj/1106314846. URL `https://doi.org/10.3150/bj/1106314846`.

Paul Gustafson. Bayesian regression modeling with interactions and smooth effects. *Journal of the American Statistical Association*, 95(451):795–806, 2000.

Trevor Hastie. Statistical learning with big data. Presentation at Data Science at Stanford Seminar, 2015.

Trevor J Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data mining, inference and prediction*. Springer Series in Statistics. Springer, 2nd edition, 2009.

Gibran Hemani, Jian Yang, Anna Vinkhuyzen, Joseph E Powell, Gonneke Willemsen, Jouke-Jan Hottenga, Abdel Abdellaoui, Massimo Mangino, Ana M Valdes, Sarah E Medland, Pamela A Madden, Andrew C Heath, Anjali K Henders, Dale R Nyholt, Eco J C. de Geus, Patrik K E. Magnusson, Erik Ingelsson, Grant W Montgomery, Timothy D Spector, Dorret I Boomsma, Nancy L Pedersen, Nicholas G Martin, and Peter M Visscher. Inference of the genetic architecture underlying bmi and height with the use of 20,240 sibling pairs. *The American Journal of Human Genetics*, 93(5):865–875, 2013. ISSN 0002-9297. doi: 10.1016/j.ajhg.2013.10.005. URL `https://doi.org/10.1016/j.ajhg.2013.10.005`.

Jennifer L Hill. Bayesian nonparametric modelling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. ISSN 00063444. URL `http://www.jstor.org/stable/2333955`.

Jared D Huling and Peter ZG Qian. Fast penalized regression and cross validation for tall data with the **oem** package. *arXiv preprint arXiv:1801.09661*, 2018.

Kosuke Imai and Marc Ratkovic. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, 7(1):443–470, 2013.

Michael J. Kane, John Emerson, and Stephen Weston. Scalable strategies for computing with massive data. *Journal of Statistical Software*, 55(14):1–19, 2013. URL `http://www.jstatsoft.org/v55/i14/`.

Hana Lango Allen, Karol Estrada, Guillaume Lettre, Sonja I. Berndt, Michael N. Weedon, Fernando Rivadeneira, et al. Hundreds of variants clustered in genomic loci and biological pathways affect human height. *Nature*, 467:832, 2010. doi: 10.1038/nature09410. URL `https://doi.org/10.1038/nature09410`.

Michael LeBlanc. An adaptive expansion method for regression. *Statistica Sinica*, 5(2):737–748, 1995.

Louis Lello, Steven G. Avery, Laurent Tellier, Ana I. Vazquez, Gustavo de los Campos, and Stephen D. H. Hsu. Accurate genomic prediction of human height. *Genetics*, 210(2):477–497, 2018. ISSN 0016-6731. doi: 10.1534/genetics.118.301267. URL `http://www.genetics.org/content/210/2/477`.

Gen Li, Xiaokang Liu, and Kun Chen. Integrative multi-view regression: Bridging group-sparse and low-rank models. *Biometrics*, 75(2):593–602, 2019.

Ruilin Li, Christopher Chang, Johanne Marie Justesen, Yosuke Tanigawa, Junyang Qian, Trevor Hastie, Manuel A. Rivas, and Robert Tibshirani. Fast lasso method for large-scale and ultrahigh-dimensional cox model with applications to uk biobank. *bioRxiv*, 2020. doi: 10.1101/2020.01.20.913194. URL `https://www.biorxiv.org/content/early/2020/01/21/2020.01.20.913194`.

Jake Lin, Rubina Tabassum, Samuli Ripatti, and Matti Pirinen. Metaphat: Detecting and decomposing multivariate associations from univariate genome-wide association statistics. *bioRxiv*, page 661421, 2019.

Luke R Lloyd-Jones, Jian Zeng, Julia Sidorenko, Loïc Yengo, Gerhard Moser, Kathryn E Kemper, Huanwei Wang, Zhili Zheng, Reedik Magi, Tonu Esko, et al. Improved polygenic prediction by bayesian multiple regression on summary statistics. *Nature Communications*, 10(1):1–11, 2019.

Adam E. Locke, Bratati Kahali, Sonja I. Berndt, Anne E. Justice, Tune H. Pers, Felix R. Day, Corey Powell, et al. Genetic studies of body mass index yield new insights for obesity biology. *Nature*, 518:197, 2015. doi: 10.1038/nature14177. URL `https://doi.org/10.1038/nature14177`.

Yen Sia Low, Blanca Gallego, and Nigam Haresh Shah. Comparing high-dimensional confounder control methods for rapid cohort studies from electronic health records. *Journal of Comparative Effectiveness Research*, 5(2):179–192, 2016.

Chongliang Luo, Jian Liang, Gen Li, Fei Wang, Changshui Zhang, Dipak K Dey, and Kun Chen. Leveraging mixed and incomplete outcomes via reduced-rank modeling. *Journal of Multivariate Analysis*, 167:378–394, 2018.

Zhuang Ma, Zongming Ma, and Tingni Sun. Adaptive estimation in two-way sparse reduced-rank regression. *Statistica Sinica*, 01 2020. doi: 10.5705/ss.202017.0073.

Zongming Ma and Tingni Sun. Adaptive sparse reduced-rank regression. *arXiv preprint arXiv:1403.1922*, 2014.

James D Malley, Jochen Kruppa, Abhijit Dasgupta, Karen G Malley, and Andreas Ziegler. Probability machines: consistent probability estimation using nonparametric learning machines. *Methods of Information in Medicine*, 51(1):74–81, 2012.

Eirini Marouli, Mariaelisa Graff, Carolina Medina-Gomez, Ken Sin Lo, Andrew R. Wood, Troels R. Kjaer, et al. Rare and low-frequency coding variants alter human adult height. *Nature*, 542:186, 2017. doi: 10.1038/nature21039. URL `https://doi.org/10.1038/nature21039`.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(Aug):2287–2322, 2010.

Nicolai Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374 – 393, 2007. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2006.12.019. URL `http://www.sciencedirect.com/science/article/pii/S0167947306004956`.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *Ann. Statist.*, 34(3):1436–1462, 06 2006. doi: 10.1214/009053606000000281. URL `https://doi.org/10.1214/009053606000000281`.

Ab Mooijaart and Jacques JF Commandeur. A general solution of the weighted orthonormal procrustes problem. *Psychometrika*, 55(4):657–663, 1990.

Guillaume Obozinski, Martin J Wainwright, Michael I Jordan, et al. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1):1–47, 2011.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1 (3):127–239, January 2014. ISSN 2167-3888. doi: 10.1561/2400000003. URL `http://dx.doi.org.stanford.idm.oclc.org/10.1561/2400000003`.

Nick Patterson, Alkes L Price, and David Reich. Population structure and eigenanalysis. *PLOS Genetics*, 2(12):1–20, 12 2006. doi: 10.1371/journal.pgen.0020190. URL `https://doi.org/10.1371/journal.pgen.0020190`.

Jie Peng, Ji Zhu, Anna Bergamaschi, Wonshik Han, Dong-Young Noh, Jonathan R Pollack, and Pei Wang. Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *The Annals of Applied Statistics*, 4(1):53, 2010.

Alkes L. Price, Nick J. Patterson, Robert M. Plenge, Michael E. Weinblatt, Nancy A. Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38:904, 2006. doi: 10.1038/ng1847. URL `https://doi.org/10.1038/ng1847`.

Florian Privé, Michael G B Blum, Hugues Aschard, and Andrey Ziyatdinov. Efficient Analysis of Large-Scale Genome-Wide Data with Two R packages: **bigstatsr** and **bigsnpr**. *Bioinformatics*, 34(16):2781–2787, 03 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty185.

Junyang Qian, Wenfei Du, Yosuke Tanigawa, Matthew Aguirre, Robert Tibshirani, Manuel A Rivas, and Trevor Hastie. A fast and flexible algorithm for solving the lasso in large-scale and ultrahigh-dimensional problems. *BioRxiv*, page 630079, 2019.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

Willi Sauerbrei, Patrick Royston, and Karina Zapien. Detecting an interaction between treatment and a continuous covariate: A comparison of two approaches. *Computational Statistics and Data Analysis*, 51(8):4054–4063, 2007.

Nigam Haresh Shah. Performing an informatics consult, May 2016. URL `http://bigdata.stanford.edu/pastevents/2016-presentations.html`. Big Data in Biomedicine Conference — Stanford Medicine.

Karri Silventoinen, Sampo Sammalisto, Markus Perola, Dorret I. Boomsma, Belinda K. Cornes, Chayna Davis, Leo Dunkel, Marlies de Lange, Jennifer R. Harris, Jacob V.B. Hjelmborg, and

et al. Heritability of adult body height: A comparative study of twin cohorts in eight countries. *Twin Research*, 6(5):399–408, 2003. doi: 10.1375/twin.6.5.399.

Timo Similä and Jarkko Tikka. Input selection and shrinkage in multiresponse linear regression. *Computational Statistics & Data Analysis*, 52(1):406–422, 2007.

Nasa Sinnott-Armstrong, Yosuke Tanigawa, David Amar, Nina J Mars, Matthew Aguirre, Guhan Ram Venkataraman, Michael Wainberg, Hanna M Ollila, James P Pirruccello, Junyang Qian, et al. Genetics of 38 blood and urine biomarkers in the uk biobank. *BioRxiv*, page 660506, 2019.

Eric Sobel, Kenneth Lange, Tong Tong Wu, Trevor Hastie, and Yi Fang Chen. Genome-Wide Association Analysis by Lasso Penalized Logistic Regression. *Bioinformatics*, 25(6):714–721, 01 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp041. URL `https://doi.org/10.1093/bioinformatics/btp041`.

Elizabeth K. Speliotes, Cristen J. Willer, Sonja I. Berndt, Keri L. Monda, Gudmar Thorleifsson, Anne U. Jackson, et al. Association analyses of 249,796 individuals reveal 18 new loci associated with body mass index. *Nature Genetics*, 42:937, 2010. doi: 10.1038/ng.686. URL `https://doi.org/10.1038/ng.686`.

Jerzy Splawa-Neyman, Dorota M Dabrowska, and Terence P Speed. On the application of probability theory to agricultural experiments. Essay on principles. Section 9. *Statistial Science*, 5(4): 465–472, 1990.

SPRINT Research Group. A randomized trial of intensive versus standard blood-pressure control. *New England Journal of Medicine*, 373(22):2103–2116, 2015.

Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M Nickerson, and Bogong Li. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research*, 10:141–158, 2009.

Matt Taddy, Matt Gardner, Liyun Chen, and David Draper. A nonparametric Bayesian analysis of heterogenous treatment effects in digital experimentation. *Journal of Business & Economic Statistics*, 34(4):661–672, 2016.

Yosuke Tanigawa, Jiehan Li, Johanne Marie Justesen, Heiko Horn, Matthew Aguirre, Christopher DeBoever, Chris Chang, Balasubramanian Narasimhan, Kasper Lage, Trevor Hastie, Chong Yon

Park, Gill Bejerano, Erik Ingelsson, and Manuel A. Rivas. Components of genetic associations across 2,138 phenotypes in the uk biobank highlight novel adipocyte biology. *bioRxiv*, 2019. doi: 10.1101/442715. URL `https://www.biorxiv.org/content/early/2019/03/19/442715`.

Lu Tian, Ash A Alizadeh, Andrew J Gentles, and Robert Tibshirani. A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association*, 109(508):1517–1532, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL `http://www.jstor.org/stable/2346178`.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 74(2):245–266, 2012. ISSN 13697412, 14679868. URL `http://www.jstor.org/stable/41430939`.

Berwin A Turlach, William N Venables, and Stephen J Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

Stephen D. Turner. **qqman**: An R package for visualizing gwas results using q-q and manhattan plots. *Journal of Open Source Software*, 3(25):731, 2018. doi: 10.21105/joss.00731.

Raja Velu and Gregory C Reinsel. *Multivariate reduced-rank regression: theory and applications*, volume 136. Springer Science & Business Media, 2013.

Thomas Viklands. Algorithms for the weighted orthogonal procrustes problem and other least squares problems. 2006.

Peter M Visscher, Sarah E Medland, Manuel A. R Ferreira, Katherine I Morley, Gu Zhu, Belinda K Cornes, Grant W Montgomery, and Nicholas G Martin. Assumption-free estimation of heritability from genome-wide identity-by-descent sharing between full siblings. *PLOS Genetics*, 2(3):1–10, 03 2006. doi: 10.1371/journal.pgen.0020041. URL `https://doi.org/10.1371/journal.pgen.0020041`.

Peter M. Visscher, Brian McEvoy, and Jian Yang. From galton to gwas: Quantitative genetics of human height. *Genetics Research*, 92(5-6):371–379, 2010. doi: 10.1017/S0016672310000571.

Peter M. Visscher, Naomi R. Wray, Qian Zhang, Pamela Sklar, Mark I. McCarthy, Matthew A. Brown, and Jian Yang. 10 years of gwas discovery: Biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017. ISSN 0002-9297. doi: 10.1016/j.ajhg.2017.06.005. URL `https://doi.org/10.1016/j.ajhg.2017.06.005`.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. https://arxiv.org/abs/1510.04342, 2015.

Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5): 2183–2202, 2009.

Jie Wang, Peter Wonka, and Jieping Ye. Lasso screening rules via dual polytope projection. *Journal of Machine Learning Research*, 16:1063–1101, 2015. URL `http://jmlr.org/papers/v16/wang15a.html`.

Herman Wold. Soft modelling by latent variables: The non-linear iterative partial least squares (nipals) approach. *Journal of Applied Probability*, 12(S1):117?142, 1975. doi: 10.1017/S0021900200047604.

Andrew R. Wood, Tonu Esko, Jian Yang, Sailaja Vedantam, Tune H. Pers, Stefan Gustafsson, et al. Defining the role of common variation in the genomic and biological architecture of adult human height. *Nature Genetics*, 46:1173, 2014. doi: 10.1038/ng.3097. URL `https://doi.org/10.1038/ng.3097`.

Marvin N Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Yu Xie, Jennie E Brand, and Ben Jann. Estimating heterogeneous treatment effects with observational data. *Sociological Methodology*, 42(1):314–347, 2012.

Jian Yang, Beben Benyamin, Brian P. McEvoy, Scott Gordon, Anjali K. Henders, Dale R. Nyholt, Pamela A. Madden, Andrew C. Heath, Nicholas G. Martin, Grant W. Montgomery, Michael E.

Goddard, and Peter M. Visscher. Common snps explain a large proportion of the heritability for human height. *Nature Genetics*, 42:565, 2010. doi: 10.1038/ng.608. URL `https://doi.org/10.1038/ng.608`.

Jian Yang, Andrew Bakshi, Zhihong Zhu, Gibran Hemani, Anna A. E. Vinkhuyzen, Sang Hong Lee, et al. Genetic variance estimation with imputed variants finds negligible missing heritability for human height and body mass index. *Nature Genetics*, 47:1114, 2015. doi: 10.1038/ng.3390. URL `https://doi.org/10.1038/ng.3390`.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. doi: 10.1111/j.1467-9868.2005.00532.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00532.x`.

Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. **Spark**: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1863103.1863113`.

Noah Zaitlen, Peter Kraft, Nick Patterson, Bogdan Pasaniuc, Gaurav Bhatia, Samuela Pollack, and Alkes L. Price. Using extended genealogy to estimate components of heritability for 23 quantitative and dichotomous traits. *PLOS Genetics*, 9(5):1–11, 05 2013. doi: 10.1371/journal.pgen.1003520. URL `https://doi.org/10.1371/journal.pgen.1003520`.

Jian Zeng, Ronald De Vlaming, Yang Wu, Matthew R Robinson, Luke R Lloyd-Jones, Loic Yengo, Chloe X Yap, Angli Xue, Julia Sidorenko, Allan F McRae, et al. Signatures of negative selection in the genetic architecture of human complex traits. *Nature Genetics*, 50(5):746–753, 2018.

Yaohui Zeng and Patrick Breheny. The **biglasso** package: A memory-and computation-efficient solver for lasso model fitting with big data in R. *arXiv preprint arXiv:1701.05936*, 2017.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

Yingqi Zhao, Donglin Zeng, A John Rush, and Michael R Kosorok. Estimating individualized

treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00503.x`.