RANKED SPARSITY: A REGULARIZATION FRAMEWORK FOR SELECTING FEATURES IN THE PRESENCE OF PRIOR INFORMATIONAL ASYMMETRY

by

Ryan Andrew Peterson

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Biostatistics in the
Graduate College of The
University of Iowa

May 2019

Thesis Supervisor: Joseph Cavanaugh, Professor of Biostatistics

ACKNOWLEDGMENTS

First, I wish to thank my incredibly thoughtful, brilliant, and generous advisor, Dr. Joe Cavanaugh. Joe, you have been instrumental in my development as a statistician, and even more so as a human being. Thank you so much for all of your support and encouragement throughout this process, even from the very beginning. On the first day I met you, you convinced me that being a statistician can in fact be as exciting as being a rock star, with an added benefit of a much longer expected life-span. Thank you for all of the great meals, rock concerts, trips, vinyl records, and wonderful memories. Having you as my advisor has been an absolute blast, and I'm looking forward to continuing our collaboration and friendship indefinitely. I could not have asked for, nor dreamt of, a better academic father.

I would also like to thank the amazing faculty of the University of Iowa's Biostatistics Department for helping me grasp and truly appreciate the field of statistics. In particular, I'd like to thank Dr. Patrick Breheny for his amazing courses that I had the fortune of taking. Patrick, your curiosity is as enlightening as it is contagious. I am very grateful to you for all of your help along the way, from my very first semester to being a member on this dissertation committee. I remember when you encouraged me to pursue the PhD after my first semester – that really meant a lot to me at the time. I think it's safe to say that you can take credit for being the first to plant the thought in my head that I may want to stay for the long haul.

I owe the rest of the thesis committee a large debt of gratitude as well; thank you for putting in the time to review this thesis and to provide useful feedback. To Dr. Phil Polgreen, thank you so much for giving me opportunities to work on projects that allowed me not only to learn about interesting scientific questions, but also to find ways to answer them creatively and to publish the results of our investigations in high-impact places. To Dr. Aaron Miller, thank you for providing me with constructive feedback for both my own research and my research assistantship, and for allowing me to take part in your fascinating projects. To Dr. Jake Oleson, thank you for all of your advice throughout my time here, and explaining that working in academia doesn't have to be a 60+ hours-per-week job (no offense, Joe!). To Dr. Ryan Miller, thank you for being an active participant in my thinking process throughout this dissertation; a lot of the important insights in this work came from our many conversations.

I am incredibly grateful to the Biostatistics Department as a whole for its role in supporting me. I certainly could not have done gotten where I am today without the Department's funding and administrative support, but it has also provided much more. The environment here is rich with mutual respect between faculty, staff, and students, and as a result I have felt incredibly valued. I am also thankful to my research assistantship team – StatEpi has been an amazing learning and collaborative experience. I consider myself extremely lucky to have been a part of such a creative, constructive, and productive research team.

Now for my peers, who helped keep me sane as a graduate student: Dr. Ben Riedle, thanks for all of the great tennis matches, I miss them and I feel as though you won the last time we played – we'd better have a rematch soon. Dr. Ryan Miller, I already thanked you as a member of the committee, but you also deserve another one here because you have been a wonderful friend and role-model. Your incessant passion and curiosity for statistics is really contagious – you must have caught something from Patrick. Soon-to-be Dr. Javier Flores – you have been an overflowing source of great memories and fun conversations. Thank you for braving the cold of Iowa; my experience here has been made immeasurably better with you here. Thank you also to my cohort: Brandon Butcher, Biyue Dai, and Anthony Rhoads, for being such great classmates; I would not have gotten through my first years here without your help and shared dedication.

To my parents, David and Sheri Peterson: thank you for being so consistently supportive, kind, loving, and inspiring. Despite constant torment from my brothers, Ricky and Chris, you somehow got me to turn out relatively well-adjusted. Thank you for teaching me discernment, wisdom, and most importantly, how to be happy. I could not have asked for better role-models. Chris and Ricky, in all seriousness, thank you for pushing me toward being my best self. Penultimately, I wish to thank my incredibly kind, generous, and supportive wife, Rebecca. Rebecca, you have completely and irreversibly improved my life. Thank you for encouraging me to fulfill even my most ambitious dreams, for being a shining beacon of fun and levity even in the deepest throes of my pursuit of this degree, and for doing this while living behind enemy lines in Hawkeye territory.

Finally, I wish to thank God for blessing me with so many great mentors, teachers, role-models, and friends, as well as for instilling in me a natural curiosity and a passion for learning. I do not live by bread alone, but by every word of God.

ABSTRACT

In this dissertation, we explore and illustrate the concept of ranked sparsity, a phenomenon that often occurs naturally in the presence of derived variables. Ranked sparsity arises in modeling applications when an expected disparity exists in the quality of information between different feature sets. Its presence can cause traditional model selection methods to fail because statisticians commonly presume that each potential parameter is equally worthy of entering into the final model – we call this principle "covariate equipoise". However, this presumption does not always hold, especially in the presence of derived variables. For instance, when all possible interactions are considered as candidate predictors, the presumption of covariate equipoise will often produce misclassified and opaque models. The sheer number of additional candidate variables grossly inflates the number of false discoveries in the interactions, resulting in unnecessarily complex and difficult-to-interpret models with many (truly spurious) interactions. We suggest a modeling strategy that requires a stronger level of evidence in order to allow certain variables (e.g. interactions) to be selected in the final model. This ranked sparsity paradigm can be implemented either with a modified Bayesian information criterion (RBIC) or with the sparsity-ranked lasso (SRL).

In chapter 1, we provide a philosophical motivation for ranked sparsity by describing situations where traditional model selection methods fail. Chapter 1 also presents some of the relevant literature, and motivates why ranked sparsity methods are necessary in the context of interactions. Finally, we introduce RBIC and SRL as possible recourses. In chapter 2, we explore the performance of SRL relative to competing methods for selecting polynomials and interactions in a series of simulations. We show that the SRL is a very attractive method because it is fast, accurate, and does not tend to inflate the number of Type I errors in the interactions. We illustrate its utility in an application to predict the survival of lung cancer patients using a set of gene expression measurements and clinical covariates, searching in particular for gene-environment interactions, which are very difficult to find in practice.

In chapter 3, we present three extensions of the SRL in very different contexts. First, we show how the method can be used to optimize for cost and prediction accuracy simulataneously when covariates have differing collection costs. In this setting, the SRL produces what we call "minimally invasive" models, i.e.

models that can easily (and cheaply) be applied to new data. Second, we investigate the use of the SRL in the context of time series regression, where we evaluate our method against several other state-of-the-art techniques in predicting the hourly number of arrivals at the Emergency Department of the University of Iowa Hospitals and Clinics. Finally, we show how the SRL can be utilized to balance model stability and model adaptivity in an application which uses a rich new source of smartphone thermometer data to predict flu incidence in real time.

PUBLIC ABSTRACT

When predicting an important outcome, often the statistician is tasked with determining which features are most predictive from a set of possibly many candidates. Some features are typically found to be linearly related to the response, while others will seem to have no correlation with the outcome, and thus do not merit inclusion in a linear model. When searching for a linear model that predicts well, most statisticians avoid exploring whether these candidate features interact with one another – and for good reason. The statistical toolbox for selecting important features relies on many assumptions that are violated when looking through large groups of unimportant features, such as interactions. However, with the advent of cross-validation and the booming popularity of black-box predictive algorithms that can capture these interactive relationships, the traditional linear model is becoming less attractive to many practitioners. This proliferation of black-box methods has negative implications for the foundational goals of science: to explain and to understand the world around us. Black-box models are by definition difficult, if not impossible, to understand and to interpret.

In this dissertation, we present new model selection tools that can equip statisticians to search for these important nonlinear and interactive relationships between candidate features and the response. Using our methods, it becomes practical in certain settings to find interpretable statistical models that can rival black-box predictive algorithms. We showcase the impressive performance of our new methods in a series of simulation studies as well as important applications, from investigating gene-environment interactions in the context of lung cancer, to predicting the number of patients who will arrive hourly at the Emergency Department of the University of Iowa Hospitals and Clinics, to predicting national incidence of the flu in real time.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
RANKED SPARSITY – A PHILOSOPHICAL MOTIVATION

1.1 Introduction

1.1.1 Background – Sparsity and Skepticism

In the model selection problem, statisticians must traditionally balance many trade-offs. We are constantly trying to balance simplicity with predictive accuracy, parsimony with complexity, sparsity with saturation, efficiency with practicality, and bias with variance. While we do have a multitude of tools at our disposal, in most realistic situations, the reliability of these instruments can change given information we cannot access. For instance, consider sparsity. A model can have many parameters, but a *sparse* model can only have a handful by definition. Does nature produce mostly sparse models? How could we possibly know this? Often times, we can understand more about a mechanism, or predict more accurately, by collecting new features. Perhaps, if we collect enough features about a generating mechanism, the remaining stochasticity would even completely vanish, allowing us to predict newly generated data *perfectly*. Putting the merits of determinism aside, statisticians often take for granted that scientists know when to stop looking for these features. However, with the advent of new computational methods and theory that helps sift through arbitrarily large sets of features, many scientists are becoming increasingly ambitious. Unfortunately, even if the computational and theoretical constraints no longer exist, a new statistical issue arises. This ambition to collect as many features as possible tends to drive down the sparsity level, like the sun with Icarus, eventually leaving the original scientific endeavor dead in the water[1].

We define the *sparsity level* as the proportion of candidate variables (a.k.a. features, covariates, or predictors) that are inactive in a given true generating model. Conversely, the *saturation level* is defined as the proportion of candidate variables that are active. The sparsity level is governed by a mix of what cannot be known about nature's true generating model and what can (sometimes) be known about the ambition of a particular scientific project.

[1] Icarus, from Greek mythology, escaped from a prison on the island of Crete by means of wings that his father, a master craftsman, constructed from feathers and wax. Despite his father warning him not to fly too high, Icarus was overcome by the joy of flight and flew too close to the sun. The heat caused his wings to melt, so he fell and drowned in (what is now called) the Icarean Sea.

It is certainly easier to *describe* sparse models; with a postulated sparse model, we can be more precise, collect less data, and improve the interpretability of our final model. And we do have tools to help us decide between sparse or saturated models: model selection criteria. Here, we define such criteria broadly as measures that are used to evaluate whether a certain predictor should be included in the model or not. These criteria typically have a penalty term, which acts as a "cost-of-admission" for each predictor (i.e. predictors are required to add a certain level of predictive efficacy in order to be admitted into the model). These popular tools come with many trade-offs of their own. Akaike's Information Criterion (AIC) (Akaike, 1974) works *efficiently* if the sample size, n, is "large" relative to the number of candidate predictors, p. It does not penalize the variables very much relative to some other penalties, which means that the models selected by AIC tend to be less sparse. It would be reasonable to conclude then that if the true model is sparse, AIC tends to do poorly (and conversely if the true model is saturated, AIC will do acceptably). The Bayesian Information Criterion (BIC) (Schwarz, 1978) works *consistently* if n is "large" relative to p. It penalizes the variables quite heavily, and the consistency property ensures that provided enough data, the selected model will be as sparse as possible. Bootstrapping and cross-validation (CV) are increasingly being used to overcome the limitations of traditional model selection criteria (most importantly the asymptotic assumptions). However, these methods can often be cumbersome to use correctly, both computationally and in terms of their interpretation.

In practice, the goal of an analysis often dictates which method/criterion is used to select a final model. If a sparse model is desired, BIC can be used. If prediction is the only concern, AIC or CV can be used. A trade-off that depends on the true generating mechanism is relegated to the background of practicality, as we recognize that we cannot realistically know whether the true model is sparse or not. Somewhere lost in that background lies another glossed-over balancing act. We are already making a presumption about the generating mechanism – a prior belief that all of the covariates are equally worthy of entering into the model. We refer to this presumption as *covariate equipoise*, and we will show how in situations where multiple levels of sparsity can be expected among covariates, the presumption of covariate equipoise can have detrimental effects on every one of these model selection tools. With this impetus, we have developed several simple solutions to this problem for many settings – settings which we refer to as *ranked sparsity*. This dissertation

will explore these techniques, which include *ranked-sparsity BIC* (RBIC), and the *sparsity-ranked lasso* (SRL).

1.1.2 A Motivating Example

The prime intuition behind ranked sparsity is that not all covariates are created equal. Why is this seemingly unconstitutional claim justifiable? Consider the following overly-simplistic example. A researcher enlists a statistician, Stacy, to predict a response y using 20 covariates and 500 observations. Stacy carefully selects which of the 20 covariates are important, looking at residual diagnostics, functional forms, etc., and at the end of the day she comes up with a model that optimally fits the data in an interpretable fashion. Then, the researcher asks Dayton, a data scientist who specializes in machine-learning, to do the same thing. Dayton uses state-of-the-art technology to train an army of randomly partitioned regression trees in a supervised learning framework. Though there is no good way to interpret Dayton's model, the predictions from the random forest are more accurate than the predictions from Stacy's model on new data. How did this happen?

One key factor is that Dayton is not necessarily concerned about treating each covariate equally. In trying to predict y, he is grabbing from an urn of practically infinite predictors (i.e. random splits of each covariate interacted with other random splits of each covariate). Seeing which ones work well, he was able to use weights, tuning parameters, and cross-valiation to cleverly aggregate the results and optimize them for prediction. Stacy only had 20 predictors in her original bin of candidates, though she did consider adding additional polynomial and spline terms. However, she didn't consider any interactions between covariates. After all, in order to check all pairwise interactions, she would have had to sift through 190 more predictors, each of which would have made the model's interpretation more nuanced. To look at 2nd order interactions would be even more preposterous; there are 1140 of them! If she were to consider all possible interactions of any order, she would have to consider $\sum_{i=1}^{20} \binom{20}{i} = 1,048,575$ candidate variables!

This phenomenon has often led statisticians like Stacy (and the authors of this work) to make the *ad hoc* claim that there should be clinically relevant evidence available beforehand to consider an interaction in a model-selection framework. Thus in the general setting, we assume *a priori* that nature's generating mechanism is completely sparse in terms of interactions. However, when we zoom out and look at science more broadly, how do scientists decide when interactions are clinically relevant, if interactions are never

considered in models?

The truth of the matter (that statisticians seem to recognize in the aforementioned *ad hoc* claim) is that *most* of the signal in a linear regression problem can be found without considering interactions, and therefore all possible interactions will add little to no signal (and an enormous amount of noise). Unfortunately, however, in making this lazy claim, statisticians have given up something very valuable: predictive efficacy. With the advent of cross-validation and machine-learning (black-box) methods that can capture interactions, data scientists have been able to make up ground in predictive modeling as a result.

Intuitively, the reason statisticians eschew interactions is that they do not expect any of their model selection tools to work well when there is a large differential in the sparsity level among candidate predictors. It stands to reason, then, that the performance of model selection criteria is directly tied to this sparsity level. In this chapter, we show that model selection criteria with higher penalties work better in situations with a high sparsity level, whereas ones with low penalties work better in situations with a high saturation level. By showing this relationship between the sparsity level and the performance of model selection criteria via a simulation study, we will justify that the optimal penalty depends greatly on the true sparsity level in the covariate space. Therefore, in situations when there is an *expected* disparity in sparsity level among candidate predictors (such as interactions vs. main effects), we must account for this by using different penalties. We provide two general frameworks for doing this in section 1.7 – first we motivate and explore RBIC, and second we motivate the SRL. In chapter 2, we explore the SRL more broadly in the context of selecting polynomials and interactions, via a simulation study and a lung cancer application. In chapter 3, we extend the SRL to ranked cost models, time-series regression, and learn-and-turn models, showcasing its versatility in very different settings.

1.2 Traditional Model Selection Criteria

Define a data generating model $G : \boldsymbol{x} \rightarrow y$, and say we are attempting to build a linear model to predict future y observations via $f(\boldsymbol{\beta}, \boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\beta}$. Model selection criteria are a useful tool for this problem, and many model selection criteria have been explored over the course of the last century. Most of them can be broadly classified as either consistent or efficient. Consistent model selection criteria aim for the goal of

selecting the most parsimonious model having the requisite structure and these include BIC, the minimum description length criterion (MDL) (Rissanen, 1978), Extended BIC (Chen and Chen, 2008), and many others. As the sample size goes to infinity, a consistent model selection criterion will certainly (with probability 1) select the correct model. Efficient model selection criteria, on the other hand, are not consistent. Instead, they attempt to minimize the error of the model when predicting the response for new data. Efficient model selection criteria include AIC (Akaike, 1974), corrected AIC (AICc) (Hurvich and Tsai, 1989), Mallow's Cp (Mallows, 1973), among many others. AICc has a "small-sample" correction that can be very useful for overcoming the shortcomings of AIC in small-sample settings.

When should we use an efficient criterion and when should we use a consistent one? When should we abandon them both? Besides the goals of a particular project, other major factors that come into play for this decision include (but are not limited to) the following: the sample size n (how much data we collected), the number of candidate predictors p (how ambitious we are), and the number of true (signal) predictors s (how lucky we were in our choices of candidate predictors). With this notation, the sparsity level is $1 - \frac{s}{p}$, and the saturation level is $\frac{s}{p}$.

One may think that it would matter whether all of the signal parameters are contained in the number of candidate predictors or not, but at least in the Gaussian response case with uncorrelated predictors, the misspecification of the model due to unobserved signal can often be subsumed into the residual variance. As such, we assume throughout this chapter that $s < p$, that is, the number of signal parameters is always less than the number of candidate predictors. We also fix n since these asymptotics are more fully explored in other work; we are primarily interested in studying what happens when varying s and p. We know that as $p \to n$, the asymptotic assumptions on which many model selection criteria are based become increasingly violated. As a result, we would expect their performance to suffer. Additionally, as $\frac{s}{p} \downarrow$ (i.e. the saturation level decreases), we may expect to see different performance in terms of efficient vs. consistent criteria. We explore this in section 1.5, but first we introduce some more basic frameworks that come into play during the model selection process.

1.3 Searching for the Correct Model

A model selection criterion can only be utilized on a model that has been fit. This is not a concern for low values of p, when all of the possible combinations of the p predictors can be fit in a model; this strategy of searching for the optimal model is called "best-subsets". However, in some cases (when p is large and/or when n is massive), it becomes impractical to fit all of these candidate models, the number of which increases with p by the order of 2^p. The more traditional solution to this dilemma has been to use a step-wise approach to search over the space of all possible models. In forward step-wise selection, the starting point is a null model. Univariate models are fit for each feature and selected from using an information criterion (IC). At each step k, models with the remaining $p - k$ features are fit, their IC values computed, and the best one selected. The process stops when all of the newly fit models increase the IC. Backwards step-wise selection is similar, but its starting point is the full model and covariates are taken out one-at-a-time until the IC stops improving. Finally, a forwards and backwards approach consists of checking not only for the addition of new features, but also the removal of features that had been included based on a previous step.

Step-wise approaches are considerably faster than best-subsets, but they still take a lot of computation time in high-dimensional or massive n settings. In these settings, the Least Absolute Shrinkage and Selection Operator (the lasso) (Tibshirani, 1996) becomes especially useful. The lasso simultanously estimates coefficients for each of the p features and selects from them, such that they are either "active" (i.e. $\hat{\beta}_j \neq 0$), or inactive $\hat{\beta}_j = 0$. The estimated nonzero coefficients do suffer from a bias that is introduced by the lasso's penalty term λ, but this bias is often warranted as it significantly cuts down on the variance of having too saturated of a model. Suppose we have features $[x_1, x_2, ..., x_p] = X_{nxp}$, and a centered response variable $\boldsymbol{y}$; some (but not all) features are related to $\boldsymbol{y}$. The lasso solution can be obtained by minimizing the following expression with respect to $\boldsymbol{\beta}$:

$$||\boldsymbol{y} - X\boldsymbol{\beta}||^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

The lasso is not a final solution to choosing a correct model, however. To make any sort of useful inference or prediction, a model must be selected along the coefficient path, which is a sequence of λ values that typically fall upon a one-dimensional grid. In other words, the tuning parameter that details how much

to shrink the nonzero coefficients, λ, must be selected somehow. This is where AIC, BIC, and CV come back into play; before this point in the process, the lasso search has been completely informed by goodness-of-fit (subject to shrinkage). In this context, all of the same properties related to model saturation hold for these criteria as was discussed in the previous section.

It is important to distinguish the difference between model search methods (like step-wise approaches and the lasso), and model selection criteria. The former is used to search for the optimal model (and in the step-wise search, is informed via model selection criteria), while the latter is used to choose the optimal model from a set of candidates. Both are typically necessary in any given situation. The lasso is in some sense a hybrid search and selection method; for every λ value along the solution path, each active coefficient is being shrunk equally (on the standardized scale). As a result, the lasso has a built-in assumption of covariate equipoise that is separate from what we discussed with model selection criteria. Therefore, a ranked sparsity tool for the lasso needs to focus not only on the model selection criteria (in selecting an optimal λ value), but also the process of searching the model space.

1.4 Related Lasso Extensions

The lasso has become very popular due to its practicality for a very large set of applications and its computational efficiency, even in lower-dimensional spaces. However, it has some drawbacks, and the literature on new lasso extensions or modifications is vast, albeit relatively young. In this section, we expound on some of these lasso extensions that we utilize elsewhere in the dissertation. We also discuss some ideas in the literature that share a similar spirit with the sparsity-ranked lasso.

Ridge Regression/Elastic Net

Ridge regression (Hoerl and Kennard, 1970) has a similar flavor to the lasso; it shrinks coefficients by a tuning parameter λ along a predefined solution path. However, ridge regression imposes its penalty on the *squared* magnitude of the coefficients (as opposed to the absolute value of the coefficients in the case of the lasso). For ridge, we minimize the following expression with respect to $\boldsymbol{\beta}$:

$$||\boldsymbol{y} - X\boldsymbol{\beta}||^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

The result of this is that the coefficients in ridge regression are never shrunk all the way to zero, and therefore no variable selection actually occurs. Still, in cases where features are highly correlated, ridge regression has the desirable feature of "sharing" the effect across these correlated features, whereas the lasso tends to pick one feature and shrink the other to zero. The lasso's choice on which covariate to select is somewhat random, informed by the size of the effects and the correlation among the features. Ridge's "share the effect" property helps to stabilize the estimates, predictions, and inferences in this correlated setting.

The elastic net (Zou and Hastie, 2005) is an attempt to get the best of both the lasso and ridge regression. Elastic net is typically parameterized with two tuning parameters, λ which constitutes overall shrinkage, and α which refers to the proportion of the penalty that corresponds to the L1 norm (e.g. $\alpha = 1$ refers to the lasso, and $\alpha = 0$ refers to ridge regression). The elastic net can often outperform the lasso, especially in settings with high correlation among features. However, the elastic net produces less parsimonious models than the lasso, and often the prediction accuracy can remain relatively unchanged between the two methods.

Adaptive Lasso

The adaptive lasso (Zou, 2006) is a two-stage approach which attempts to penalize each covariate differently in a data-driven fashion to get around the lasso's issue of bias. A "first glance" is taken at the data and used to inform the modeling framework about which predictors look to be most important right away. This first glance consists of an initial estimator for all of the model's coefficients, which directly informs the weight of the L1 penalty for that predictor in the lasso expression. With the adaptive lasso, we minimize the following with respect to $\boldsymbol{\beta}$:

$$||\boldsymbol{y} - X\boldsymbol{\beta}||^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|$$

The addition of the w_j is the only difference between this formulation and the lasso. The adaptive lasso fixes the w_j values in a way that corresponds to its "first glance". For example, one popular method is to set $w_j = |\tilde{\beta}_j|^{-1}$, where $\tilde{\boldsymbol{\beta}}$ is the ordinary least squares estimate of $\boldsymbol{\beta}$ (the ridge estimator is another option). The adaptive lasso has been shown to have the *oracle* property. This means that the adaptive lasso estimator for $\boldsymbol{\beta}$ asymptotically has the same properties of the true model's maximum likelihood estimators. The lasso does

not have this oracle property.

The sparsity-ranked lasso methods that we introduce do share some similarities to the adaptive lasso. In fact, we can describe the SRL in the form of the adaptive lasso solution (though the SRL does not qualify as a variant of the adaptive lasso necessarily). In chapter 3 in particular, one of our extensions, SRLPAC, has very close ties with the adaptive lasso that we will discuss. For the most part, however, we do not take this "first glance" approach to informing the weights. Rather, we take an intuitive, context-specific approach.

IPF Lasso

The Integrative Lasso with Penalty Factors (IPF-lasso) (Boulesteix et al., 2017) is perhaps the closest in spirit to a ranked-sparsity method; the idea is that each group of covariates should have their own estimated penalty. Originally developed in the context of building models with multiple types of -omics feature sets, the IPF-lasso simply creates a new tuning parameter for each covariate group k, estimating λ_k using a grid search and cross-validation. The solution is similar to the adaptive lasso, except that $w_j = \lambda_k$ for all coefficients j in group k. This method, while technically possible for any number of groups, becomes computationally difficult when multiple groups are considered; usually the resolution of tuning parameters must be sacrificed in order to satisfactorily search the tuning parameter space. In some sense, this flexibility can be beneficial or harmful depending on the situation. We explore the similarities of the sparsity-ranked lasso to the IPF lasso further in the discussion of this chapter.

Group Lasso/Sparse Group Lasso

The group lasso (Yuan and Lin, 2006) extends the lasso for situations when covariates are "grouped", and we expect either all or none of the within-group coefficients to be equal to zero. In other words, we expect that groups are either fully saturated or fully sparse. A selection procedure based on such a conjecture may be advisable with categorical predictors, where multiple columns of a design matrix represent categories of the same predictor. If the ordinary lasso is applied in this setting, a clustering would result centered at the baseline category, and the lasso is thus sensitive to the choice of baseline category. We are primarily interested in what happens when there are multiple groups within which there are different levels of sparsity, so the group lasso is not very applicable.

However, an extension of this group lasso idea, the sparse group lasso (SGL) (Friedman, Hastie and

Tibshirani, 2010; Simon et al., 2013), is closely related to our field of inquiry. Essentially, the SGL "mixes" the lasso and the group lasso in a similar fashion that elastic net mixes the lasso and ridge regression. SGL has two tuning parameters: λ, which denotes the overall penalty, and α, which denotes the proportion of the penalty that should be applied to the coefficients directly (as opposed to the group as a whole). For example, $\alpha = 1$ corresponds to the ordinary lasso, whereas $\alpha = 0$ corresponds to the group lasso. The result (for $\alpha \in (0, 1)$) is that coefficients are penalized similarly to other coefficients within their group; if a coefficient is in a group with other important variables (i.e. those with large effects), it is penalized less than a coefficient in a group without other large coefficients. The similarities of the SGL and the SRL are explored in the discussion.

Nonconvex Regularization

The Smoothly Clipped Absolute Deviations penalty (SCAD) (Fan and Li, 2001) and the Minimax Concave Penalty (MCP) (Zhang, 2010) were designed to allow the bias of the lasso to go to zero as the magnitude of the coefficient increases. Because the penalty function with respect to each coefficient is not convex (nor is it concave), these both fall under the umbrella of *nonconvex regularization*. These methods both employ an additional tuning parameter, γ, which can tune how quickly the bias in the coefficients tapers off as the magnitude of the coefficient increases. This additional tuning parameter allows these methods to have the oracle property. MCP and SCAD tend to work better than the lasso in settings with high signal relative to the amount of noise, but they can suffer from instability as a result of the lack of global convexity. The lasso is a special case of SCAD and MCP (when $\gamma = \infty$), and as γ is lowered, the instability increases. Typical choices for γ that have been shown to be effective in many situations are $\gamma = 4$ for SCAD and $\gamma = 3$ for MCP. Both of these methods are explored further in relation to ranked sparsity in chapter 2.

1.5 Simulation Study – Model Selection and Sparsity Levels

1.5.1 Simulation Setup

In this simulation, we explore the finite-sample behaviors of AIC (as an exemplary efficient criterion) and BIC (as an exemplary consistent criterion) for various sparsity/saturation levels. We wish to show the following for a fixed n.

1) A higher saturation level requires an information criterion with a lower penalty. Specifically:

   a) a model selected with BIC will outperform AIC in a setting with a low saturation level, and

   b) a model selected with AIC will outperform BIC in a setting with a high saturation level.

2) As the number of candidate variables p approaches n, this disparity increases.

3) Cross-validation (CV) can circumvent this issue by selecting the optimal penalty based on extra-sample performance.

For this simulation, we take the sample size to be 500, the number of total candidate variables $p \in \{250, 350, 450, 490\}$, the number of signal variables $s|p = p * (.01, .02, .04, .1, .25, .5, .75, .95, .99)$, and we repeat the simulation 100 times. For every p and sparsity level, we generate p independent standard normal variables which represent the covariate space X, an n x p matrix of independent standard normal random variables. We then generate $\boldsymbol{y} = X\boldsymbol{\beta} + N(\mathbf{0}, 10^2)$, where $\boldsymbol{\beta}^T = (\mathbf{1}_s^T, \mathbf{0}_{p-s}^T)$. In other words there are s signal variables and $p-s$ null variables for each simulation. Since the number of non-zero coefficients is changing, the signal-to-noise ratio (SNR $= \boldsymbol{\beta}^T Var(X)\boldsymbol{\beta}/\sigma^2$) changes as well for different sparsity levels. This is by design; we argue that truly sparse settings are typically settings where the SNR is low, whereas truly saturated settings indicate a higher SNR. However, we will also investigate what happens when the SNR is fixed. See Table 1.1 for more information. We henceforth denote the saturation level as ω, where $\omega = s/p$.

In order to search for the optimal model, we use the lasso and select the tuning parameter λ according to AIC, BIC, and CV. We utilize the lasso as a model selection tool (as opposed to a step-wise procedure) for several reasons. First and foremost, the lasso can be quickly and efficiently run for many candidate variables simultaneously; step-wise methods take much more time and are somewhat sensitive to selection direction (forward vs. backward). Additionally, the use of the lasso procedure lends itself well for extension into the realm of ranked sparsity.

In order to determine the quality of the selected fit by each criterion, we generate $10*n$ new observations and responses, calculating each model's root-mean-squared prediction error (RMSPE) on the newly generated data. We use the RMSPE (as defined in section 3.3.4) because it is a practical model selection metric, and because we know the generating distribution. While it does not directly consider whether the model has the right covariates or not, model misspecification (either under- or over-specification) will still inflate the

Table 1.1: Number of true signal variables and effective signal-to-noise ratios (SNR) for simulations investigating performance of model selection criteria in tuning lasso models with different "true" saturation levels s/p (sample size fixed at 500).

	$p = 250$	$p = 350$	$p = 450$	$p = 490$
Number of Signal Variables				
$\omega = 0.01$	2	3	4	4
$\omega = 0.02$	5	7	9	9
$\omega = 0.04$	10	14	18	19
$\omega = 0.1$	25	35	45	49
$\omega = 0.25$	62	87	112	122
$\omega = 0.5$	125	175	225	245
$\omega = 0.75$	187	262	337	367
$\omega = 0.95$	237	332	427	465
$\omega = 0.99$	247	346	445	485
Effective SNR				
$\omega = 0.01$	0.02	0.03	0.04	0.04
$\omega = 0.02$	0.05	0.07	0.09	0.09
$\omega = 0.04$	0.1	0.14	0.18	0.19
$\omega = 0.1$	0.25	0.35	0.45	0.49
$\omega = 0.25$	0.62	0.87	1.12	1.22
$\omega = 0.5$	1.25	1.75	2.25	2.45
$\omega = 0.75$	1.87	2.62	3.37	3.67
$\omega = 0.95$	2.37	3.32	4.27	4.65
$\omega = 0.99$	2.47	3.46	4.45	4.85

RMSPE in cases where the model is badly specified. To showcase our points 1), 2), and 3) above, we plot the mean RMSPE across 100 simulations by the saturation level in a series of plots.

1.5.2 Results

In Figure 1.1, we observe that as expected, there is a large discrepancy in the predictive quality of the model fits selected by AIC and BIC depending on the sparsity level. When the saturation level is low, and n is high relative to p, AIC and BIC performed relatively similarly. As $p \to n$, however, AIC begins to perform quite poorly relative to BIC, especially in more sparse settings. BIC outperforms AIC in sparse settings, especially when p is approaching the sample size n. However, AIC would be preferable to BIC in the saturated model setting provided that n is decently higher than p. In all cases, (10-fold) cross-validation performed optimally at selecting the optimal λ value. Thus, we have shown in this example that the optimal penalty depends, to an extent at least, on the sparsity level (which we typically cannot know ahead of time).

Again, in this simulation setting, the SNR varies with the saturation level, but a similar simulation was performed that holds the SNR fixed at 1 for each simulation (by setting $\sigma = 1$ and varying the magnitude of the coefficients for different sparsity levels). These supplemental simulations still showed AIC perform increasingly poorly as $p \to n$. However, when the saturation level became higher than about .1, BIC was curiously able to do relatively well compared to AIC, and the disparity in criterion performance was more difficult to see as a result. More investigation into the properties of BIC when the SNR changes is merited, but is beyond the scope of this work. Now, we turn our attention to cases where we anticipate there being a different sparsity level across all of the covariates.

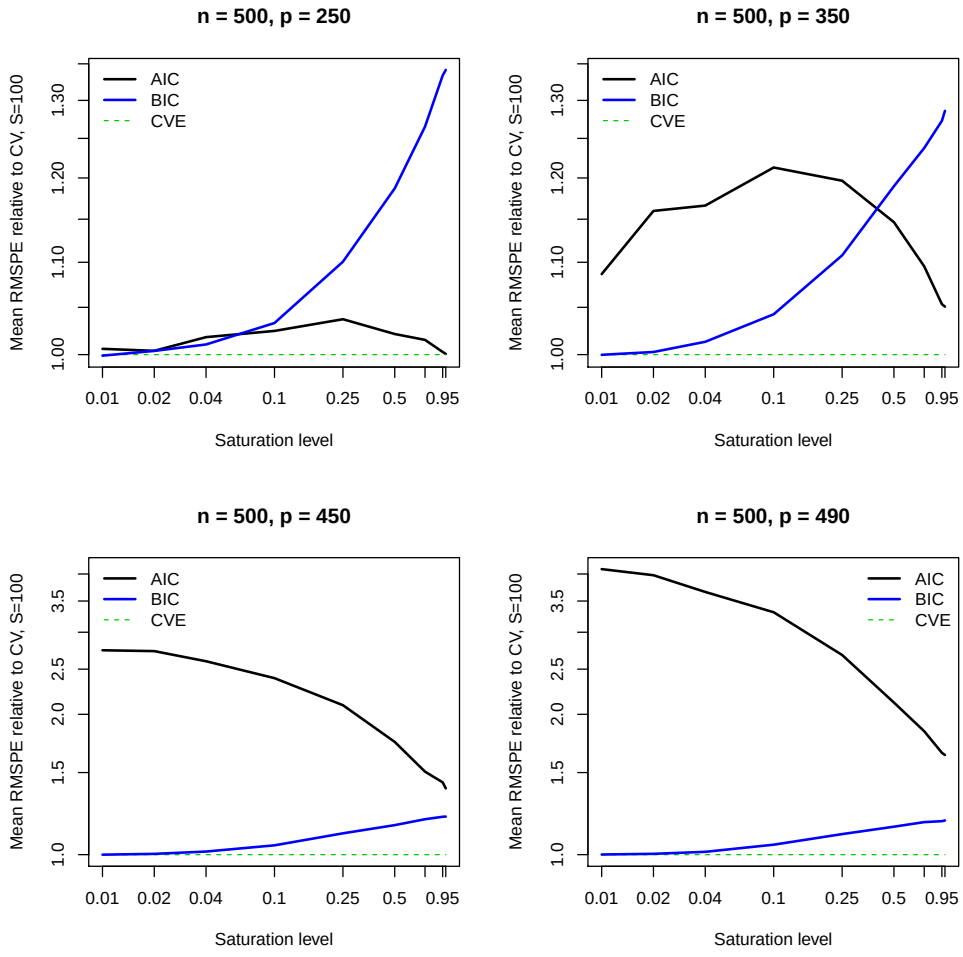

Figure 1.1: Predictive accuracy when using AIC, BIC, or CV for different values of p and different saturation levels.

1.6 Ranked Sparsity with Interactions

It is quite clear that AIC and BIC perform very differently for different sparsity levels when the sample size is held constant, and that this difference increases as the number of candidate predictors approaches the sample size. While this disparity can be avoided by using cross-validation to select the optimal penalty instead of AIC/BIC, it still indicates that the performance of a penalty is highly dependent on the sparsity level. This provides motivation in the context of ranked sparsity, where we expect groups of covariates to have different sparsity levels.

Take the context of selecting from all possible pairwise interactions. We define "main effects" to refer to linear coefficients in a model, and "interaction effects" to refer to coefficients on the product of covariates that correspond to the main effects. We can show that given a saturation level in the main effects, the maximum saturation level attainable for the interaction effects is limited by hierarchy assumptions about the true generating model. Model hierarchy is typically broken down into "strong", e.g. $y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} * x_{2i}$, "weak", e.g. $y_i = \beta_0 + \beta_1 x_{1i} + \beta_3 x_{1i} * x_{2i}$, and "anti" hierarchical models, e.g. $y_i = \beta_0 + \beta_3 x_{1i} * x_{2i}$. For instance, if there are only 3 active variables of 30 possible main effects, then strong hierarchy would dictate that in the generating model, only $\binom{3}{2} = 3$ signal variables can exist in the interaction set. Under weak hierarchy in the generating model, then this quantity is limited to $\sum_{j=1}^{3} 30 - j = 97$ active interactions. In either case, the number of signals in the interaction set is bounded by the number of signal variables in the main effects, as are their sparsity levels.

In fact, by this logic, a strongly hierarchical generating model ensures that the saturation level in pairwise interactions is necessarily less than the saturation level in the main effects. Let M refer to the number of signals in the main effects of p possible predictors, and let I refer to the number of signals in the interactions of $\binom{p}{2}$ possible interactions. Also, let ω_m and ω_I refer to the saturation levels in the main effects and the interactions, respectively. Assume that $p \geq 2$ and that $p\omega_m \in \mathbb{Z}$. By definition, we have that

$$\begin{aligned} M &= p\omega_m \\ I &= \binom{p}{2}\omega_I \\ \Rightarrow \omega_I &= \frac{I}{\binom{p}{2}} \end{aligned} \tag{1.1}$$

Under strong hierarchy, the signals in the interactions must be combinations of the signals in the main effects, which implies

$$I \leq \binom{p\omega_m}{2} \tag{1.2}$$

Then, by (1.1) and (1.2),

$$\max \omega_I = \frac{\binom{p\omega_m}{2}}{\binom{p}{2}} = \frac{\omega_m(p\omega_m - 1)}{p-1}$$

This upper bound is plotted as a function of ω_m in Figure 1.2 (left), and it is evident from the figure that $\max \omega_I < \omega_m \ \forall \ \omega_m \in (0,1)$. Note that using L'Hôpital's rule, we know that $\lim_{p\to\infty} \max \omega_I = \omega_m^2$. Under weak hierarchy, instead of the expression in (2), we have

$$I \leq \sum_{j=1}^{m} p - j = \frac{1}{2} p\omega_m(2p - p\omega_m - 1)$$

The corresponding bound on the saturation level of the interactions then becomes

$$\begin{aligned} \max \omega_I &= \frac{\frac{1}{2}p\omega_m(2p - p\omega_m - 1)}{\binom{p}{2}} \\ &= \frac{\omega_m(2p - p\omega_m - 1)}{p-1} \end{aligned}$$

This upper bound is shown in the right plot of Figure 1.2. We see that in the weakly hierarchical case, the maximum saturation level in the interactions is still bounded by the signals in the main effects, though it is not restricted to be less than the saturation level in the main effects. If the generating mechanism is not hierarchical, there is no necessary bound set on ω_I by ω_m.

The reason for going through these derivations is to show that in this setting, there is a very defensible reason to believe that the sparsity level in one set of covariates (interaction effects) is going to be less than the sparsity level in another group of covariates (main effects). As a result, it becomes necessary to account for this somehow in the model selection process; we cannot apply the same penalty to both main effects and interactions and expect optimal performance.

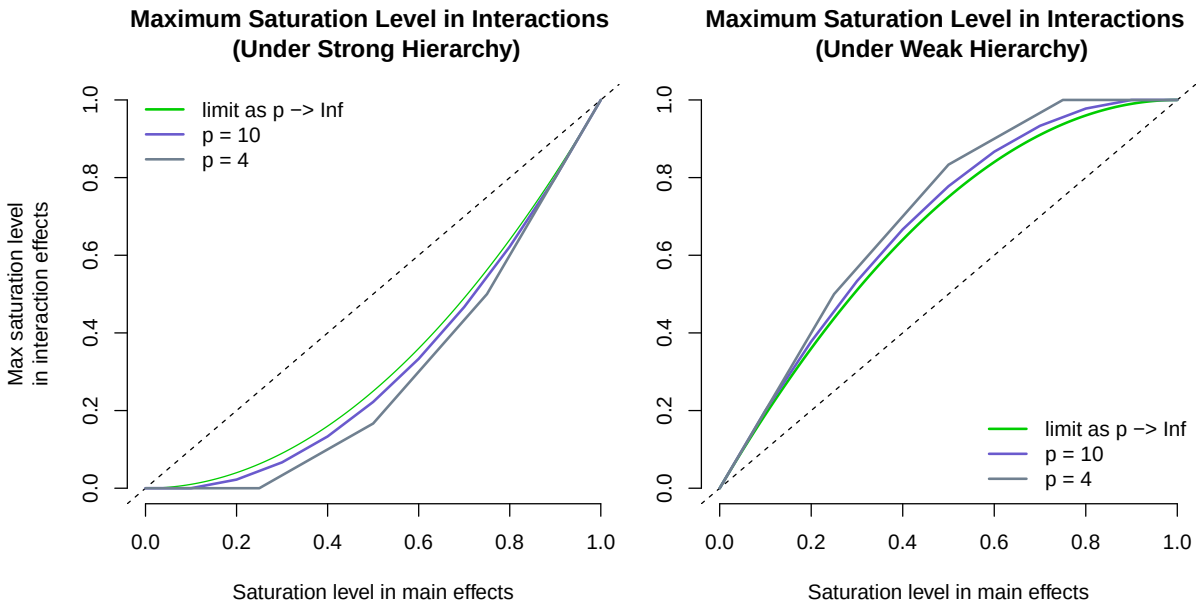

Figure 1.2: In either the strong (left) or weak (right) hierarchy setting, the maximum saturation level in the interaction effects is bounded by the saturation level in the main effects.

1.7 Ranked Sparsity Methods

1.7.1 Ranked Sparsity for Low-Dimensional Spaces

The Extended Bayesian Information Criterion (EBIC) corrects BIC for cases when $p > n$ (Chen and Chen, 2008). In their paper, the authors argue that EBIC is useful for especially sparse models when "BIC is too liberal". They suggest that cross-validation is often too liberal, and the EBIC can improve upon it as well. One of the standard conditions in BIC is that the number of total parameters p is fixed; they show that adding a term that increases with p is crucial in ensuring consistency when p increases to infinity with n. Finally, they show how if $p > \sqrt{n}$, BIC is likely to be inconsistent.

EBIC has a nice intuitive motivation. Consider doing best-subsets selection with 10 covariates; one may think that a good way of selecting the final model would be to choose the one that has the lowest BIC. However, BIC puts a uniform prior on all candidate models, which can become problematic, since there are many more models with 5 covariates than models with 1 or 2 covariates. The prior distribution for model size

is thus centered among values near to $p/2$ (see Figure 1.3). As p increases, this prior distribution becomes concentrated quickly around $p/2$. Consequently, the prior distribution on the saturation level becomes dense around $\omega = 0.5$.

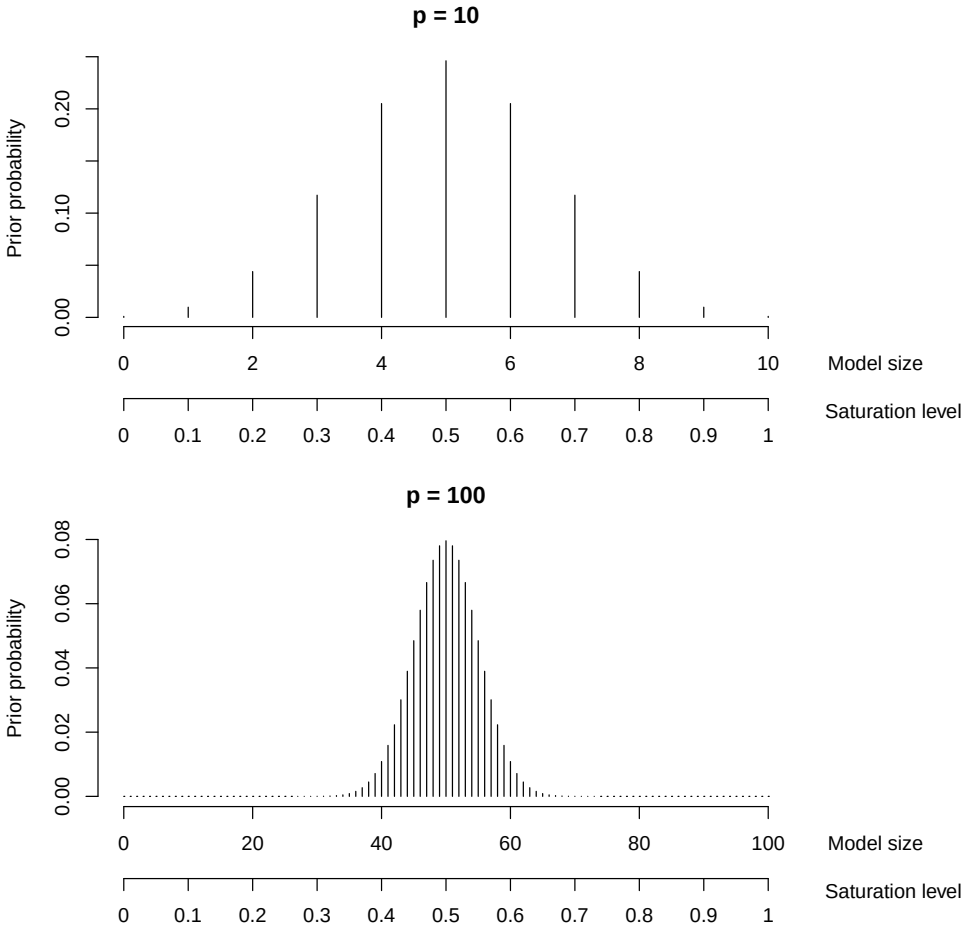

Figure 1.3: The prior instituted by BIC on the marginal distribution of model size (and, consequently, the saturation level) for p = 10 (top) and p = 100 (bottom).

EBIC corrects for this prior imbalance by incorporating an additional term to BIC that penalizes a model in accordance with the number of candidate models of that size. Let p refer to the number of covariates in a model selection problem, let n be the sample size and let m be the dimension of a single candidate model

with parameter vector $\boldsymbol{\beta}$.

$$\begin{aligned}\text{EBIC} &= -2\log l(\hat{\boldsymbol{\beta}}) + m\log n + 2\gamma\log\binom{p}{m} \\ &= \text{BIC} + 2\gamma\log\binom{p}{m}\end{aligned}$$

A good way of conceptualizing this additional term for EBIC is with the statistician's favorite metaphor: balls and urns. In any variable selection problem, each covariate can be thought of as a ball in an urn. Each model is then a random draw of m balls from that urn. There are $\binom{p}{m}$ ways of selecting m balls from an urn with p total, which is the reason this combination appears in the formula for EBIC.

Different values of γ lead to important special cases, and can be visualized in Figure 1.4. If $\gamma = 0$, EBIC becomes the original BIC. If $\gamma = 1$, EBIC is consistent when $p = O(n^\kappa)$ for any $\kappa \geq 0$ not depending on n. In other words, $\gamma = 1$ yields a consistent EBIC as long as p does not increase exponentially with n. This latter case puts a uniform prior on the marginal distribution of model size. However, setting $\gamma = 1$ can be too stringent; depending on the values of n and p, one may be able to lower γ and still achieve consistency. For example, if $\gamma = 0.5$, EBIC is consistent when $\kappa < 1$, and we know that BIC (i.e. $\gamma = 0$) is consistent when $p < \sqrt{n}$.

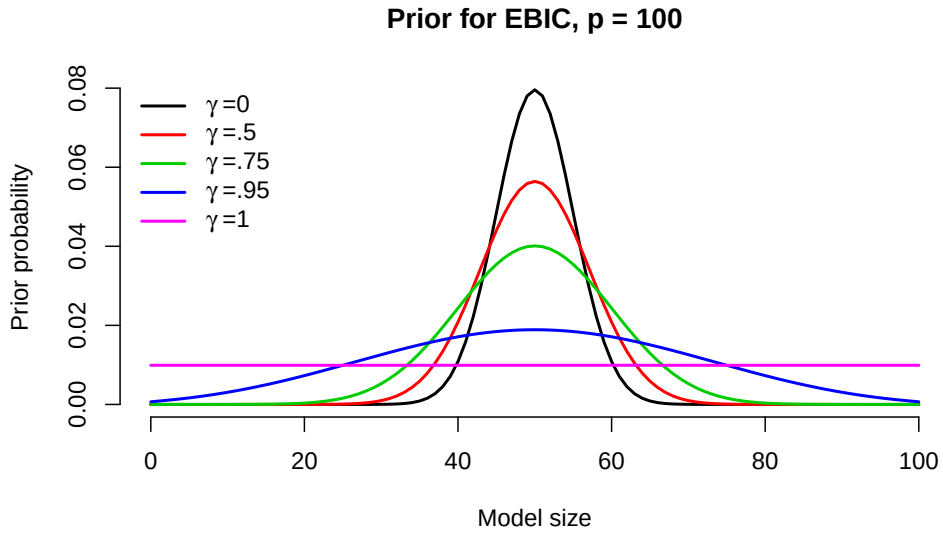

Figure 1.4: The γ parameter for EBIC.

In practice, the exact choice for γ is still somewhat unclear. In the original paper, the authors suggest

that one way of choosing γ is to solve for κ from $p = n^\kappa$ and then set $\gamma = 1 - 1/2\kappa$.

$$\gamma \overset{\text{set}}{=} 1 - \frac{\log n}{2 \log p} = \frac{\log(p/\sqrt{n})}{\log p}$$

Since it is possible for the term above to be less than zero (if $p < \sqrt{n}$), we use

$$\gamma \overset{\text{set}}{=} \max\left(\frac{\log(p/\sqrt{n})}{\log p}, 0\right)$$

So, the extra term for EBIC in this case is

$$\frac{2 \log(p/\sqrt{n}) \log \binom{p}{m}}{\log p} \mathbf{1}(p \geq \sqrt{n})$$

In Figure 1.5, we show how this additional term for EBIC changes with various other values, including p, n, and m.

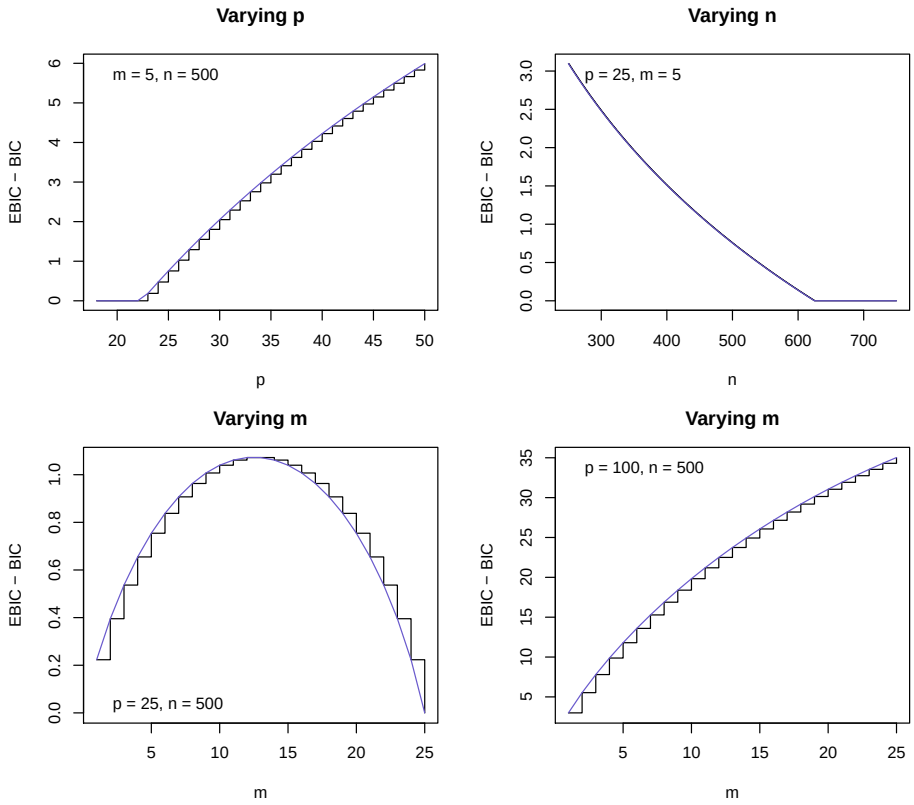

Figure 1.5: EBIC's extra term. Higher values along the y-axis indicate a higher penalty; specifically, the additional penalty on top of BIC. Since p is discrete, the black lines represent the actual difference, and the blue lines represent interpolated values.

We use this same "balls and urns" metaphor to intuitively motivate a ranked sparsity extension of BIC (RBIC). Instead of one urn with p covariates, we postulate that we have multiple (say, K) urns, each with p_k covariates. For instance, we could have one urn containing our main effects, and another urn containing interactions of those main effects.

The probabilities involved are only slightly more complex. For a candidate model with m_k parameters of p_k possible for all $k \in \{1, 2, ..., K\}$,

$$\begin{aligned} \text{RBIC} &= -2 \log l(\hat{\boldsymbol{\beta}}) + \sum_k m_k \log n + \sum_{k=1}^{K} 2\gamma_k \log \binom{p_k}{m_k} \\ &= \text{BIC} + \sum_{k=1}^{K} 2\gamma_k \log \binom{p_k}{m_k} \quad , \quad 0 \leq \gamma_k \leq 1 \; \forall \; k \end{aligned}$$

This formulation of RBIC effectively penalizes a covariate's coefficient differently depending on which covariate subspace (or urn) it comes from. In the case of pairwise interactions, the penalty (when $p > 2$) is higher for interaction effects than for main effects. We have explored various options for the γ_k, and simulations suggest that setting all of the γ_k to the same value as suggested for EBIC performs comparably well. This simulation study is presented in the next section. It is worth noting that setting $\gamma_k = 0 \; \forall \; k$ yields the original BIC, and if $K = 1$, RBIC is identical to EBIC.

RBIC (and EBIC) can be implemented and understood best in a best-subsets selection framework, but it is also amenable to any step-wise selection framework. However, several advancements in the computational efficiency of IC-based model searching, such as those implemented by the R packages `leaps` (Lumley and Miller, 2017) and `bestglm` (McLeod and Xu, 2018), are unavailable for RBIC. These packages are able to take advantage of the covariate equipoise presumption; all models with m parameters have the exact same penalty ($m \log n$ for BIC, or $2m$ for AIC). With RBIC, each covariate may be penalized differently, so not all models with m parameters have the same penalty.

1.7.2 RBIC in Action

In order to study how RBIC compares to its competitors, we perform a simulation study. Data in this simulation study was generated under the Friedman model (Friedman, 1991), shown below.

$$y = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + N(0, 3^2)$$

In addition to these 5 "signal" terms, 5 noise variables were generated as well (all independent uniform random variables). Given 80 observations, models were fit using forwards-and-backwards step-wise regression using AIC, BIC, EBIC, and RBIC. For RBIC, we present two possibilities for the γ parameter, which distinguishes $RBIC_1$ and $RBIC_2$.

$$RBIC_1 = BIC + 2\sum_{j=1}^{K} \frac{\log(p/\sqrt{n})}{\log p} \mathbf{1}\left(p > \sqrt{n}\right) \log \binom{p_j}{m_j}$$

$$RBIC_2 = BIC + 2\sum_{k=1}^{K} \frac{\log\left[\left(\sum_{j=1}^{k} p_j\right)/\sqrt{n}\right]}{\log\left(\sum_{j=1}^{k} p_j\right)} \mathbf{1}\left(\sum_{j=1}^{k} p_j > \sqrt{n}\right) \log \binom{p_k}{m_k}$$

$RBIC_1$ refers to RBIC with the γ parameter selected in the same fashion as in $EBIC$, and $RBIC_2$ refers to RBIC with a γ_k parameter that cumulatively penalizes the groups more depending on the ordering of the groups. Note that the ordering of the covariate groups of size p_k matters a lot in the formulation whenever $\sum_{j=1}^{k} p_j$ is used in the penalty. For the purposes of this simulation, the ordering is as follows:

$$X_{nxp} = \{X_{\text{nx10}}, X^{\odot 2}_{\text{nx10}}, X^{\odot 3}_{\text{nx10}}, \mathbb{X}^1_{\text{nx45}}, \mathbb{X}^2_{\text{nx120}}\}$$

where the notation $X^{\odot k}$ refers to the element-wise squared ($k = 2$) or cubed ($k = 3$) terms in X, $\mathbb{X}^1$ refers to the element-wise products of each column to all of the others (all possible pairwise interactions), and $\mathbb{X}^2$ refers to all possible 3-way interactions. Thus, $p_1 = p_2 = p_3 = 10$, $p_4 = \binom{p_1}{2} = 45$, and $p_5 = \binom{p_1}{3} = 120$.

We investigate how RBIC performs in the linear regression setting (no candidate polynomials or interactions), the quadratic regression setting (squared terms and pairwise interactions), and the cubic regression setting (all terms in X above). Note that in the linear regression setting, $K = 1$, so RBIC and EBIC are identical. To quantify model performance, we calculate the RMSPE on newly generated data, as well as the number of false negatives (NFN), the number of false positives (NFP), and the coefficient-wise selection percentage. Based on the form of the Friedman model, the covariates that we assessed to have legitimate nonzero coefficients are x_1, x_2, x_4, x_5, x_1^2, x_2^2, x_3^2, and the pairwise interaction $x_1 * x_2$; all other covariates, if selected, were considered false positives. Finally, in order to test for differences between EBIC, $RBIC_1$, and $RBIC_2$ in terms of fitting quality, we use a gamma (log link) generalized linear model (GLM) for RMSPE, and Poisson (log link) GLMs for NFN and NFP. The results of 1,500 simulations are presented

in the following figures and tables.

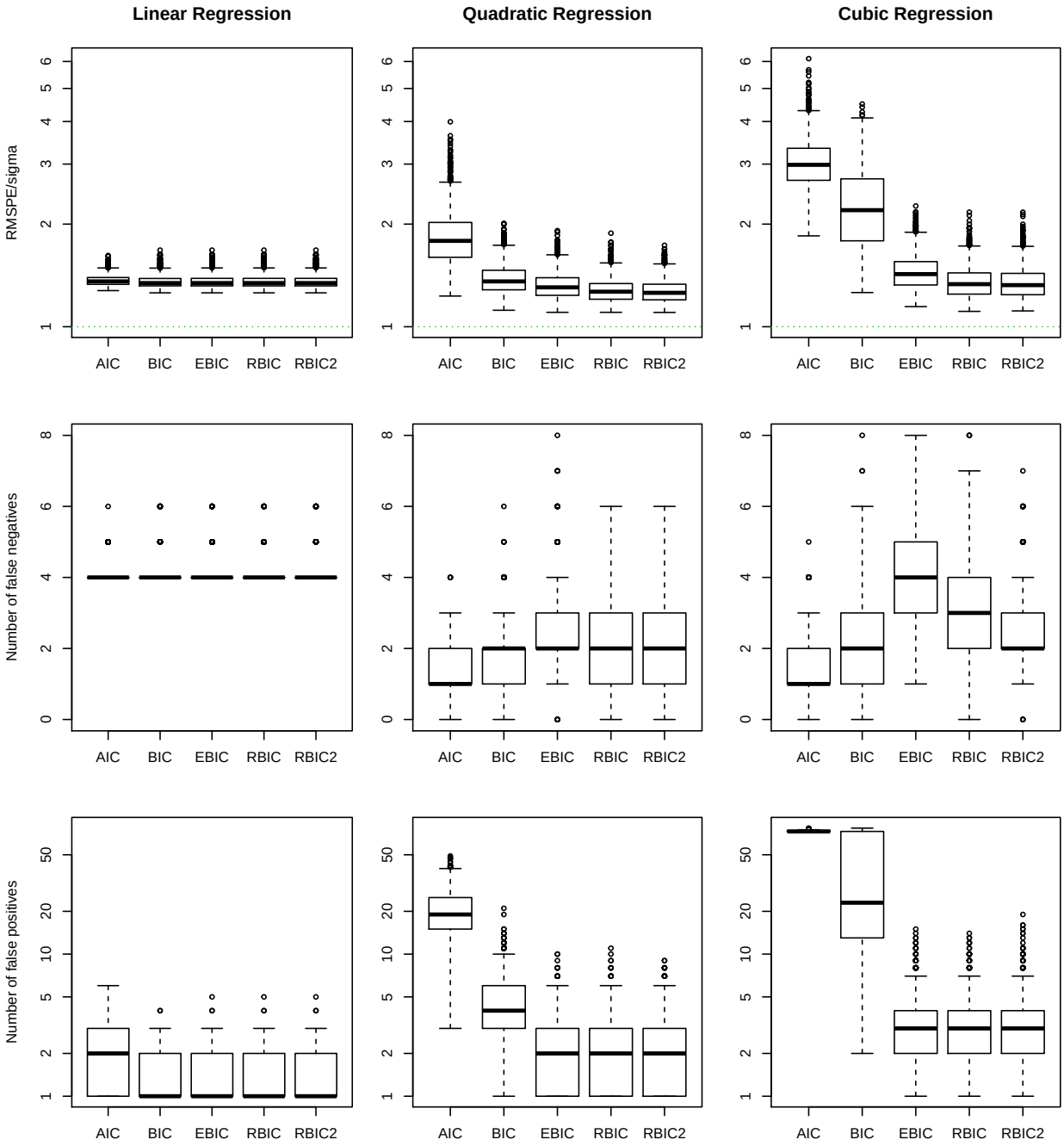

Figure 1.6: Results for 1,500 simulations under the Friedman generating model, where models were selected with fowards-and-backwards step-wise selection under either AIC, BIC, EBIC, and RBIC.

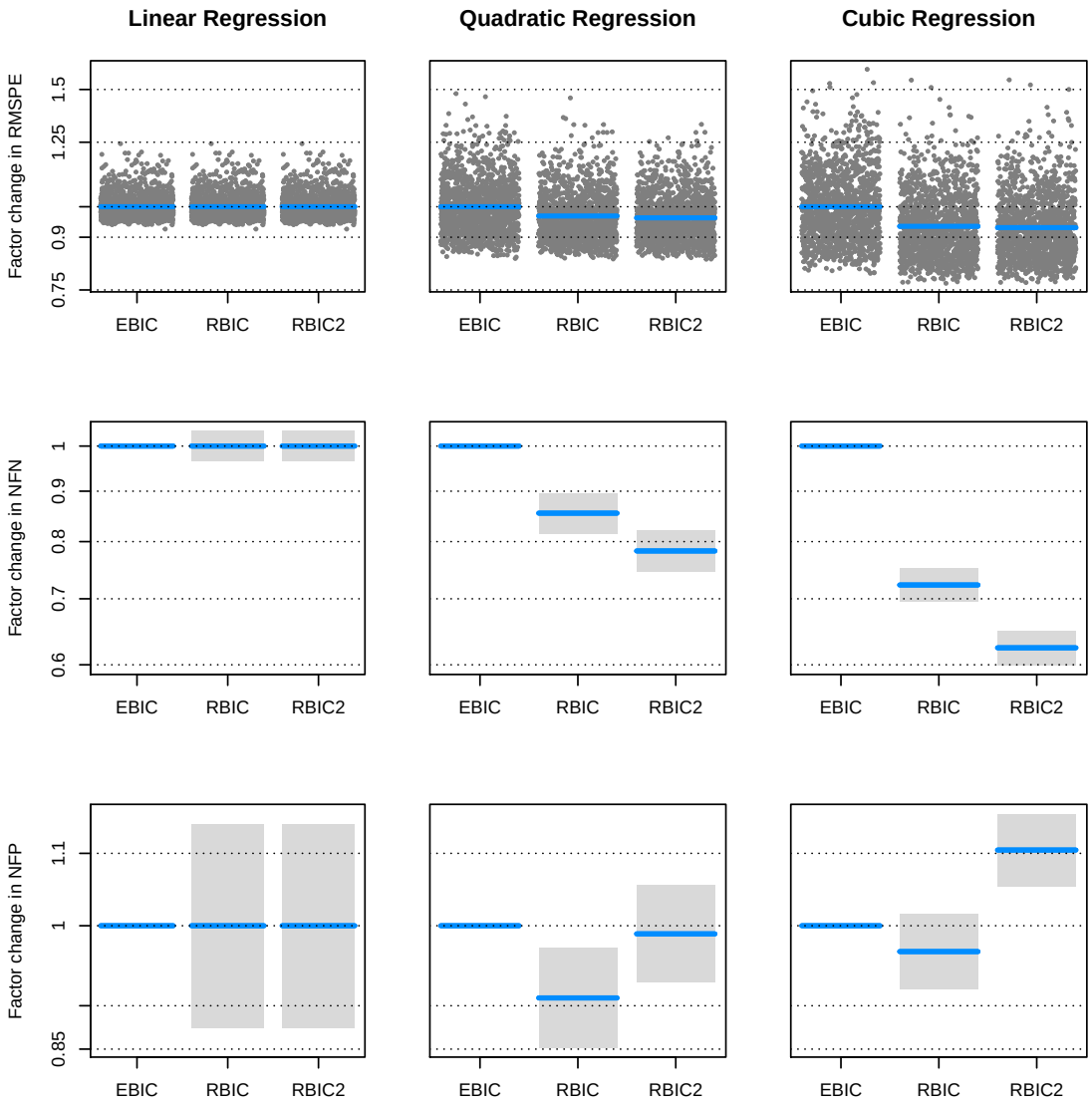

Figure 1.7: Results from GLM fits to RMSPE, NFN, and NFP values with each simulation as a data point. Points correspond to the exponentiated linear predictors taking EBIC as a baseline, and are omitted in the NFP and NFN plots due to the discreteness of these outcomes and to clarify the inferences. For NFN and NFP, only the factor change and 95% CI are plotted. AIC and BIC are also omitted here for clarity (they performed badly in the quadratic and cubic regression setting). The y-axis refers to the factor change in the expected outcome relative to the expected value of EBIC.

These simulation results show that both AIC and BIC work well only in the linear regression setting, but fail badly in the quadratic or cubic regression setting (in terms of prediction and the number of false positives). This is likely due to the violation in asymptotic assumptions as the number of candidate predictors is higher than the sample size for these settings. EBIC performed similarly to RBIC in terms of prediction,

Table 1.2: Mean across simulations of the root-mean-squared prediction error (RMSPE), the number of false negatives (NFN), and the number of false positives (NFP) for models selected via step-wise selection using varying information criteria. Linear, quadratic, and cubic refer to the included candidate predictors.

	AIC	BIC	EBIC	$RBIC_1$	$RBIC_2$
Mean RMSPE					
Linear	4.10	4.08	4.08	4.08	4.08
Quadratic	5.57	4.15	3.98	3.85	3.83
Cubic	9.18	6.84	4.37	4.08	4.06
Mean NFN					
Linear	4.04	4.18	4.18	4.18	4.18
Quadratic	1.24	1.87	2.56	2.19	2.00
Cubic	1.49	2.40	4.07	2.95	2.54
Mean NFP					
Linear	1.09	0.28	0.28	0.28	0.28
Quadratic	19.08	3.79	1.25	1.14	1.24
Cubic	72.49	39.53	2.13	2.06	2.35

though RBIC saw a small (but statistically significant) gain in predictive efficacy relative to EBIC for both quadratic and cubic regression. This seems to have been accomplished by a slightly lower number of false negatives, while there was little change in the number of false positives between EBIC and RBIC.

$RBIC_1$ and $RBIC_2$ performed quite similarly in terms of prediction. The main difference seems to be a slightly different balance between false negatives and false positives; $RBIC_2$ had more false positives than $RBIC_1$ but had fewer false negatives. This is likely due to the "ranking" preference that is set up in the γ_k for $RBIC_2$. In Table 1.3, we observe that $RBIC_2$ exhibits higher power for the signals of "lesser" order, compared to $RBIC_1$. Neither method was able to completely account for the fact that the added cubic polynomials were largely noise; the RMSPE went up when going from quadratic to cubic regression for all of the methods. This suggests that while RBIC may be a good alternative to other model selection methods, it only mediates the issue and does not completely account for the fact that adding extraneous polynomials and interactions makes model selection more difficult.

1.7.3 Ranked Sparsity for High-Dimensional Spaces via the Lasso

We have already used the lasso and seen how it can be tuned to varying degrees of effectiveness with different model selection criteria. We now motivate an extension to the lasso that can be utilized to account

Table 1.3: Selection percentage for signal variables across simulations and information criteria. Linear, quadratic, and cubic refer to the predictors that were included as candidates. "Avg T1 error" refers to the mean selection percentage across noise variables.

	AIC	BIC	EBIC	$RBIC_1$	$RBIC_2$
Linear Regression					
x_1	99.7	97.6	97.5	97.5	97.5
x_2	99.5	97.3	97.3	97.3	97.3
x_4	100.0	100.0	100.0	100.0	100.0
x_5	97.2	87.2	86.9	86.9	86.9
Avg T1 error	18.1	4.7	4.7	4.7	4.7
Quadratic Regression					
x_1	99.0	98.0	94.7	97.9	98.5
x_2	98.7	98.4	95.2	97.7	98.8
x_4	100.0	100.0	99.8	99.9	100.0
x_5	94.9	92.2	80.6	90.5	92.4
x_1^2	77.1	62.7	45.9	54.0	59.3
x_2^2	75.0	60.9	43.3	52.6	57.5
x_3^2	95.1	91.3	80.6	86.2	90.7
$x_1 * x_2$	36.6	9.8	4.3	2.6	2.7
Avg T1 error	33.5	6.6	2.2	2.0	2.2
Cubic Regression					
x_1	100.0	88.7	72.6	84.3	88.6
x_2	98.7	88.7	72.5	85.0	89.7
x_4	96.8	91.8	86.0	89.2	94.0
x_5	84.5	70.6	48.7	74.6	79.0
x_1^2	69.0	56.4	24.1	44.9	51.7
x_2^2	70.1	56.6	23.4	45.0	53.0
x_3^2	90.7	83.8	63.1	80.7	87.5
$x_1 * x_2$	41.3	23.3	2.1	1.7	2.2
Avg T1 error	38.8	21.1	1.1	1.1	1.3

for ranked sparsity; we call this extension the sparsity-ranked lasso (SRL). A more thorough exploration of the SRL is presented in chapters 2 and 3, but we provide the motivation for the SRL in this section.

Recall the lasso solution presented in section 1.3. It is well-known that this solution has a Bayesian interpretation. If each $\beta_j \sim \text{Laplace}(0, \lambda)$, then the *mode of the joint posterior distribution* represents the lasso solution (Tibshirani, 1996). To help visualize this idea, we created a shiny app that is available at https://ph-shiny.iowa.uiowa.edu/rpterson/shiny_vis1/. Notably, the mode of the joint posterior is the same as the lasso solution for a given λ value. As the sample size increases, the likelihood becomes more concentrated and contributes more information to the posterior, eventually pulling the mode off of zero. As λ is increased, the balance of information shifts toward the Laplace prior, and the mode gets pulled (or

"snapped") to zero. These zero-centered independent Laplace priors form the following joint prior density:

$$\pi(\boldsymbol{\beta}) = \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda|\beta_j|}$$

As a brief aside, we turn to the concept of Fisher information. Fisher information is invoked in likelihood theory to describe the behavior of maximum likelihood estimators, but at its heart, Fisher information is a concept that can quantify the structural characteristics of any joint density. Jeffreys, for instance, derived his famous "noninformative" prior based on the concept of the Fisher information of a *prior* density. For $W \sim f(w|\lambda)$ where $\lambda \in \Lambda$ is scalar and $\lambda \to \log f(w|\lambda)$ is twice differentiable in λ for every w, the model Fisher information at any λ is defined to be

$$I(\lambda) = E_{W|\lambda}\left[-\frac{\partial^2}{\partial\lambda^2} \log f(W|\lambda)\right]$$

With these concepts in mind, consider partitioning the covariate space X from before into K groups, such that

$$X = [A_1, A_2, ..., A_k, ..., A_K]$$

Let p_k refer to the column dimension of $A_k \; \forall \; k$, and let β_j^k refer to a particular β_j in covariate group k, then the prior for $\boldsymbol{\beta}$ undergoes a purely cosmetic change and becomes

$$\pi(\boldsymbol{\beta}|\lambda) \propto \prod_{k=1}^{K} \prod_{j=1}^{p_k} \lambda e^{-\lambda|\beta_j^k|}$$

Take λ to be a parameter. If we think of all of the β_j^k as random variables (which they are *a priori*), it becomes straightforward to find the Fisher information in this prior density.

$$\begin{aligned}
\log \pi(\boldsymbol{\beta}|\lambda) &= c + \sum_{k=1}^{K} \sum_{j=1}^{p_k} \left(\log \lambda - \lambda|\beta_j^k|\right) \\
&= c + \sum_{k=1}^{K} p_k \log \lambda - \lambda \sum_{k=1}^{K} \sum_{j=1}^{p_k} |\beta_j^k| \\
\frac{\partial}{\partial\lambda} \log \pi(\boldsymbol{\beta}|\lambda) &= \frac{1}{\lambda} \sum_{k=1}^{K} p_k - \sum_{k=1}^{K} \sum_{j=1}^{p_k} |\beta_j^k| \\
\frac{\partial^2}{\partial\lambda^2} \log \pi(\boldsymbol{\beta}|\lambda) &= -\frac{1}{\lambda^2} \sum_{k=1}^{K} p_k
\end{aligned}$$

So, the Fisher information in the joint prior is

$$I(\lambda) = E_{X|\lambda}\left[-\frac{\partial^2}{\partial\lambda^2}\log\pi(\boldsymbol{\beta}|\lambda)\right] = \frac{1}{\lambda^2}\sum_{k=1}^{K} p_k$$

This increases with the dimension of the parameter spaces (for any $\lambda > 0$). In other words, if $p_1 = 10$, and $p_2 = 100$, by default the covariate group A_1 contributes a tenth of the prior information as does the covariate group A_2 for any λ. In many situations, this weighting scheme may not be desired. What if we were to set the prior information for each parameter group to be equal? By slightly modifying (scaling) the prior distribution, we can accomplish this desired weighting. Replacing λ with $\lambda_k = \lambda\sqrt{p_k}$, we have independent and identical Laplace prior distributions for each β_j^k with mean 0 and variance $\frac{1}{\lambda^2 p_k}$.

$$\pi(\boldsymbol{\beta}|\lambda) \propto \prod_{k=1}^{K}\prod_{j=1}^{p_k}\sqrt{p_k}\lambda\exp\left\{-\lambda\sqrt{p_k}|\beta_j^k|\right\}$$

And, letting $\lambda_k \stackrel{\text{def}}{=} \sqrt{p_k}\lambda$, we have

$$\begin{aligned}\log\pi(\boldsymbol{\beta}|\lambda) &= c + \sum_{k=1}^{K}\sum_{j=1}^{p_k}\left(\log\lambda_k - \lambda_k|\beta_j^k|\right)\\ &= c + \sum_{k=1}^{K} p_k\log\lambda_k - \sum_{k=1}^{K}\lambda_k\sum_{j=1}^{p_k}|\beta_j^k|\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial\lambda_k}\log\pi(\boldsymbol{\beta}|\lambda_k) &= \frac{p_k}{\lambda_k} - \sum_{j=1}^{p_k}|\beta_j^k| \quad \forall\ k\\ \frac{\partial^2}{\partial\lambda_k^2}\log\pi(\boldsymbol{\beta}|\lambda_k) &= -\frac{p_k}{\lambda_k^2} \qquad\qquad \forall\ k\end{aligned}$$

So the model's Fisher information contained in the prior for covariates in group k (i.e., the combined information contained in the priors on β_j^k) is

$$I(\lambda_k) = \frac{p_k}{\lambda_k^2} = \frac{p_k}{\lambda^2 p_k} = \frac{1}{\lambda^2}\ \forall\ k$$

This indicates that the contributed information of each group of covariates to the joint prior is equal, and the combined information across all of the β_j is

$$\sum_{k=1}^{K} I(\lambda_k) = \frac{K}{\lambda^2}$$

Therefore by simply scaling all of the penalties by the square-root of their respective group size, we can

achieve a ranking in the sparsity that treats covariate *groups* equally as opposed to the covariates themselves. After going through this motivation for the SRL, we found that this formulation of the lasso is related to the (sparse) group lasso, which features a similar scaling by $\sqrt{p_k}$ (Yuan and Lin, 2006; Simon et al., 2013). A discussion of the role of $\sqrt{p_k}$ scaling in connection to the uniformly most powerful invariant test in the group lasso setting appears in Simon and Tibshirani (2012). Further exploration of the SRL is presented in the next two chapters of this dissertation.

Connection to multiple comparisons

One way of considering the issue that arises with the lasso under covariate groups of different sizes is to examine what happens with the expected number of false discoveries (FD) in each group. In order to estimate this quantity in the lasso problem, we can use a normal approximation (under certain conditions) (Breheny, 2018; Miller and Breheny, 2019) :

$$E(FD) \approx 2p\Phi\left(\frac{-n\lambda}{\hat{\sigma}}\right) = p\alpha$$

Here $\hat{\sigma}$ is an estimate of the residual standard deviation, α is the false discovery rate, and Φ is the standard normal CDF. In words, given a value for λ and an estimate of the residual variance, the expected number of false discoveries, $E(FD)$ can be approximated with $p\alpha$. In our "partitioning" formulation, if we were to run an ordinary lasso, the number of false discoveries in each group k is $E(FD_k) = p_k\alpha$, which means we expect the number of variables falsely selected in a model to be proportional to the size of each group. In a sense, the *rate* of false discoveries in each group (i.e. $E(FD_k)/p_k$) is constrained to be the same by default. However, it may be of interest to restrict the *number* of false discoveries within each group to be approximately equal instead of this rate. When examining pairwise interactions, for instance, we would expect there to be many more false discoveries in the interactions than in the main effects.

In Figure 1.8, we showcase this disparity. Note that as p_1 (the number of main effects) increases, the number of false discoveries in the interactions increases much more quickly than the number in the main effects. The blue dotted line signifies the ratio of these two quantities. Interpreting this, we find that with 11 candidate main effects, we expect there to be 5 false discoveries in the pairwise interactions for every 1 false discovery in the main effects. With 21 main effects, this increases to 10 interaction false discoveries for every

main effect false discovery. For 51 and 101 candidate main effects, this ratio is 25 and 50, respectively. These ratios are simply the ratio of the sizes of the covariate groups, and they do not depend on α.

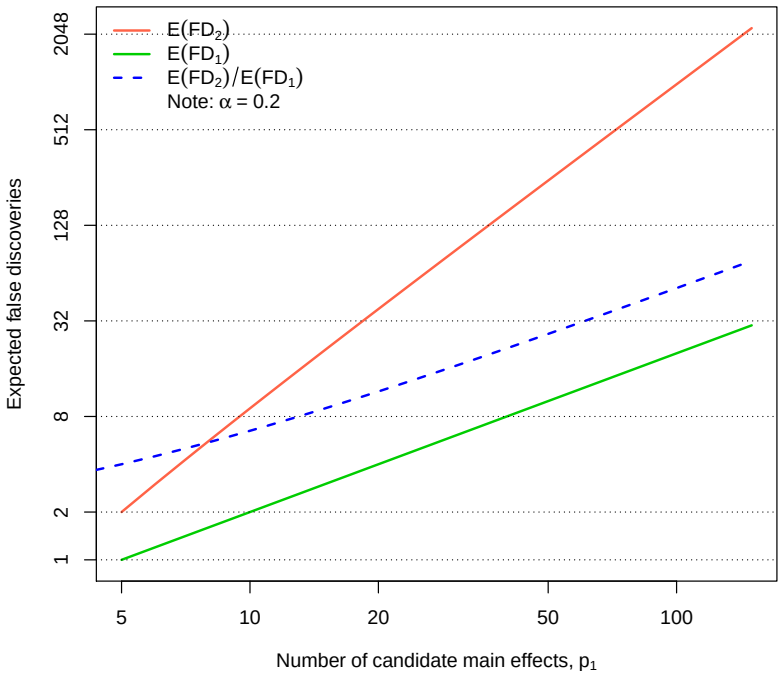

Figure 1.8: Disparity in the number of false discoveries between pairwise interactions and the number of candidate main effects.

In theory, it may be possible to set up the weights for the lasso problem to attempt to control the expected number of false discoveries to be proportional (or equal) among covariate groups. However, this extension quickly becomes difficult practically and computationally, since the weights must change with both λ and σ^2. We believe that these weights would also be highly sensitive to the quality of the estimate of σ. After investigating the feasibility of such an approach, it became clear that the scaled-Laplace prior information formulation of ranked sparsity via the lasso would be a much more practical and useful approach.

In some sense, this formulation of the ranked sparsity problem mirrors the logic behind correcting for multiple comparisons in the first place. The Bonferroni correction, for instance, emphasizes that there be a greater degree of evidence to show statistical significance when there are multiple tests being performed. For secondary subgroup tests in a clinical trial, these adjustments can be important, but traditionally the

primary endpoint does not get adjusted. Ranked sparsity methods operationalize this form of "primary" and "secondary" thinking.

1.8 Discussion

We have shown that the cost-of-admission required for each predictor in any given model selection problem should be connected to the expected sparsity level. While this quantity is typically unknown, we do often know ahead of time whether there should be an ordering to the sparsity levels of certain predictors. In these settings, we have shown that it is important to account for this expected ordering, and we have offered two novel methods to accomplish this: RBIC and the SRL. RBIC in particular outperformed both EBIC and BIC in the selection of variables in quadratic and cubic regression. This, along with the exploration of the SRL in chapter 2, shows that both of these methods can be effectively used to select from derived variables such as interactions and polynomials.

As we mentioned in section 1.4, the IPF-lasso is a similar idea to the SRL in that each coefficient's group is penalized differently depending on its group, and we can estimate all of these specific penalties using CV. However, in practice this solution does not work well for the situation where groups have substantial differences in group size, such as the case of looking for pairwise interactions. Boulesteix et al. (2017) advise to investigate penalty factors within each group as $(1, 2^\gamma)$ for a sequence of positive and negative integers γ (in the 2-group case). This does not yield an optimal solution if candidate 2^γ values are not near $\sqrt{\binom{p}{2}}$, which is the value which assumes equal information among the main effects and the interaction effects. By not incorporating the dimension of each group in the tuning of the model, the tuning parameter selection is not as precise as it could be (as it is in the proportional information setting). Searching for the optimal penalty factor for the IPF lasso is like a blindfolded archer, who despite being pointed in the right direction, has no clue how far away her target is. So, she shoots several arrows, some near some far. Most would argue that this archer is more effective without the blindfold.

For example, in one application, Boulesteix et al. (2017) investigate combining clinical data (11 features) with microarray gene expression measurements (22,283 features) in order to predict the survival of patients in a breast cancer study. Their CV procedure for finding optimal penalties showed optimal penalty

factor for the genetic expression data to be 32, penalizing the molecular data much more than the clinical data (in fact, there were no selected gene expressions in this model). With the SRL, we do not need to go through the CV procedure to find this value. In order for the gene-expression measurements to contribute the same amount of prior information as the clinical features, this value should be $\sqrt{22,283/11} \approx 45$. Therefore, the penalty factor of 32 is likely too low. A penalty factor of 45 on the genetic expression features would still select none of those features, in that sense the model would be the same. However, with a penalty factor of only 32, the coefficients on the clinical data are shrunk more than they should be, and this is likely to be decreasing the power to detect the clinical effects as well as the predictive accuracy of the final model.

Another related lasso extension that merits discussion is the sparse group lasso (SGL). The solution to the SGL is found by minimizing the following with respect to $\boldsymbol{\beta}$:

$$||\boldsymbol{y} - X\boldsymbol{\beta}||^2 + \alpha\lambda \sum_{k=1}^{K} \sum_{j=1}^{p_k} |\beta_j^k| + (1-\alpha)\lambda \sum_{k=1}^{K} \sqrt{p_k} ||\boldsymbol{\beta}^k||$$

As mentioned in section 1.4, SGL bears a resemblence to the SRL; notice the similarity to the SRL in terms of the factor multiple of the group-level penalty, $\sqrt{p_k}$. However, the SGL can yield quite different results to the SRL in practice. This formulation shrinks the magnitude of the entire vector $\boldsymbol{\beta}^k$ within each group. The SRL, on the other hand, penalizes each coefficient in some sense independently from the others in its group. For example, if the magnitude of the first coefficient in group one is large, i.e. $\hat{\beta}_1^1 >> 0$, the SRL would not induce any effect on the magnitude of $\hat{\beta}_2^1$. The SGL still penalizes each variable separately in its first penalty, but its second penalty is on the group-level magnitude. In our toy example, a large $\hat{\beta}_1^1$ coefficient would thus relax the penalty on $\hat{\beta}_2^1$ to an extent. See Figure 1 in Friedman et al. (2010) for a good visualization of this. A more detailed comparison of the performance of the SGL to the SRL is left for future work, but one benefit to the SRL that is already evident is its lack of the need for an additional tuning parameter, as $\gamma = .5$ is a good choice for most circumstances.

CHAPTER 2
MODEL SELECTION IN THE PRESENCE OF DERIVED FEATURES WITH THE SPARSITY-RANKED LASSO

2.1 Introduction

2.1.1 Background

In the ever-growing, ever-changing field of model selection and machine learning, "black-box" predictive models are becoming increasingly popular (and increasingly opaque). When one's exclusive desire is predictive accuracy, these difficult-to-interpret models are often worth a certain lack of understanding. However, overly complex predictive contexts are not generally compatible with the traditional aim of science: to describe and to understand the world in which we live. With their growing popularity, black-box models are starting to be applied in situations where description should be the primary goal. Worse, in some circumstances, there is little regard for the consideration that more transparent models could produce similar prediction results.

We argue that one of the stipulations of Occam's Razor is that transparent models should be preferred to opaque models, and that the benefits reaped from choosing a black-box model should be weighed against the interpretative costs of a lack of scientific understanding. However, before we can proffer a method to accomplish this goal, we must first answer the question – why do black-box methods outperform transparent models in prediction? The answer is difficult because these black-box methods are diverse, as are the situational considerations that make a particular method perform better or worse. Broadly speaking, the benefits of black-box methods can be roughly explored by investigating situations where transparent linear models fail. We will focus in particular on the issue of bias caused by model misspecification.

Say we have a set of data made up of a response of interest $\boldsymbol{y}$ and a set of covariates (or predictors) X, some of which are related to $\boldsymbol{y}$ while others are not. One can envision many ways of fitting an optimal predictive model to $\boldsymbol{y}$, but a popular method (if transparency is a goal) is to fit linear models based on all possible subsets of covariates to select the best one on the basis of an information criterion. This method is somewhat limited to lower-dimensional settings, because the number of candidate models increases combinatorically with the dimension of X. However, in recent times the Least Absolute Shrinkage and

Selection Operator (the lasso) has changed the landscape surrounding the problem of identifying a suitable predictive model (Tibshirani, 1996). With the lasso and its many extensions, it is possible to have an extremely high-dimensional covariate space and still end up with a relatively well-fit, interpretable model. Unfortunately, in either of these settings, black-box methods exist that can be applied and would probably improve the model if the true generating model has informative interactions among covariates and/or nonlinear relationships between a covariate and the response. This is because none of the linear candidate models can capture the model's complex interaction/polynomial terms; all of the candidate transparent models are misspecified.

One potential solution, given the power and flexibility of the lasso, would be simply to add "derived variables" of X, such as interactions and polynomials, into a new (and potentially very large) model design matrix X^*. The lasso *can* simultaneously select and estimate interactions and polynomials that are important even in this ultra-high dimensional setting. However, in this chapter, we will show that this method is severely flawed. We will present the concept of *ranked sparsity*, along with a regularization method that can correct for known problems with the aforementioned approach: the sparsity-ranked lasso (SRL). We perform a simulation study to investigate the performance of the SRL compared to competing methods, and we apply the SRL in a high-dimensional setting of gene-environment interaction selection in the context of lung cancer. Finally, we discuss the strengths and weaknesses of the SRL relative to other strategies that have been proposed.

2.1.2 Ranked Sparsity Intuition

Ranked sparsity, which we will also refer to as *ranked skepticism*, is a philosophical framework that challenges the traditional implementation of Occam's Razor in the context of variable selection. In the words of Einstein, the maxim stipulates that "everything must be made as simple as possible, but not simpler." This is a noble goal, but some questions arise: how do we know when a model is as simple as it should be? How should we measure simplicity in the first place? Specifically, we wish to challenge the ubiquitous answers to these questions in the field of model selection, which rely on a presumption that we call *covariate equipoise*: the belief that all covariates are equally likely to enter into a model. To illustrate this idea, say we are trying to find a well-fit model to predict an outcome $\boldsymbol{y}$ using a set of covariates, including age, weight, and height.

Of the candidate models below, which is "simpler"?

$$y_i = \beta_0 + \beta_1 \text{Age}_i + \beta_2 \text{Weight}_i + \beta_3 \text{Age}_i * \text{Weight}_i \tag{2.1}$$

$$y_i = \beta_0 + \beta_1 \text{Age}_i + \beta_2 \text{Weight}_i + \beta_3 \text{Height}_i \tag{2.2}$$

Virtually all variable selection tools assume these two models to be equally simple. Due to the presumption of covariate equipoise, simplicity is equated to *parsimony*, and is measured only by the number of parameters in the model (which is 4 for both models here). However, any statistician, when choosing between these two models, would recognize that model (2.2) is an order of magnitude easier to understand and communicate than model (2.1). We argue that simplicity should not only be tied to parsimony, it should also be tied to the transparency and the interpretability of a proposed model. This is the primary motivation for the sparsity-ranked lasso, which we describe in detail in the next section, after a brief review of other methods for selecting from polynomials and interactions.

2.1.3 Polynomial Selection and Smoothing

Polynomial models of the form $E(Y) = \beta_0 + \sum_{j=1}^{k} \beta_j x^j$ have traditionally been utilized when there is a nonlinear relationship to be expected between x and y. While effective in many circumstances, they are becoming less popular due to the increasing computational practicality of other, more robust "smoothing" techniques such as splines and nonparametric regression. With a polynomial model, the first concern is how to choose the maximum order k. While traditional model selection tools such as AIC and BIC can be used in this regard to some degree of effectiveness, the problem becomes intractable in higher dimensional settings. Further, if there is even a slight amount of skew in the covariate data, high-order polynomials can become highly correlated with lower order polynomials, which can lead to variance inflation. Worse still, in the skewed covariate setting, higher order polynomials tend to contain highly influential points, which makes the model selection (and the selection of the optimal k) more difficult.

Splines, on the other hand, are quite robust to the distribution of the covariate. They can accomplish this by having "knots", which are points along x that serve as junctions where piece-wise polynomials must connect. A similar model selection issue exists with splines, namely in how many effective degrees of freedom

to spend in the fitting of the spline. Again, information criteria or cross-validation can be utilized to balance goodness of fit with parsimony effectively. This is done by default in the `mgcv` package with the `gam()` and `s()` functions; for more information, refer to Wood (2017) and the documentation of `mgcv`.

Kernel-based methods are also commonly utilized for smoothing. These methods are nonparametric, and are more computationally intensive than the previous methods. One particular variety of kernel smoothing is called local regression, of which locally estimated scatter plot smoothing (LOESS) is a special case. In the LOESS framework, subsets of the data are selected via a nearest-neighbor (NN) algorithm and polynomial fits (typically of order 1 or 2) are fit using weighted least squares, with the weights provided by NN. LOESS is flexible, and only requires the specification of two parameters: the smoothing parameter that controls the "wiggliness", and the degree of the local polynomials. Another kernel-based approach is to use a moving window along the range of x to estimate the expected value within that window, where each point within the window gets either equal weight (referred to as the "boxcar" approach), or Gaussian weights. This approach only requires specification of the weighting scheme and the bandwidth of the moving window.

While each of these methods has its own strengths and weaknesses in different settings, some benefits are unique to the traditional polynomial fit. First, it has interpretative simplicity; especially for small k, a polynomial regression can be well described, understood, and communicated without the use of a figure. Secondly, in high-dimensional settings, marginal polynomial fits can become more attractive than kernel-based methods due to their computational efficiency. Splines, on the other hand, can be efficiently implemented in sparse high-dimensional settings with the group lasso via SPAM (Ravikumar et al., 2009), although if the relationships of interest are truly linear, SPAM will tend to overfit them.

In selecting the optimal order of polynomials, most model selection methods, as we have seen, treat each parameter equally *a priori*. Such a premise is not necessarily advisable in this setting, however. Realistically, and especially in high-dimensional settings, statisticians have a natural predisposition to consider a linear relationship most likely, and their skepticism increases with each added polynomial. This is partially due to the fact that lower k models are more interpretable, and less prone to overfitting. Our ranked-sparsity paradigm can operationalize this predisposition, building it into the model selection process. In section 2.3, we compare the effectiveness of traditional polynomial fits using ranked-sparsity methods to smoothers in a

series of simulations.

2.1.4 Interaction Selection and the Marginality Principle

In the context of interactions, many authors have written about the *marginality principle*, also known as model hierarchy or model heredity. Model hierarchy is typically broken down into "strong", e.g. $y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} * x_{2i}$; "weak", e.g. $y_i = \beta_0 + \beta_1 x_{1i} + \beta_3 x_{1i} * x_{2i}$; and "anti" hierarchical models, e.g. $y_i = \beta_0 + \beta_3 x_{1i} * x_{2i}$. There are several methods that purport to do well in selecting and estimating interactions under the weak and/or strong hierarchy constraint. The hierNet approach (Bien, Taylor and Tibshirani, 2013) is well-suited for low-dimensional problems due to its computational complexity. A similar regularization-based method, glinternet (Lim and Hastie, 2015), has been shown to be as effective as hierNet in selecting interactions, but able to execute the fitting and selection 10-10000 times faster. The "strong heredity interaction model" (SHIM) approach is similar to the hierNet approach; it extends the lasso to select interaction terms while under a strong hierarchy constraint. SHIM also adds an adaptive lasso element to achieve the oracle property (Choi, Li and Zhu, 2010), and uses an IPF-lasso style approach of tuning the penalty for the interactions separately from the main effects. SHIM thus has an additional tuning parameter to cross-validate over. Finally, a newer approach is called "regularization under marginality principle" (RAMP) (Hao, Feng and Zhang, 2018). RAMP is a two-stage regularization approach that is useful for settings where the storage of the interaction model matrix is an issue. By having a first-stage screening via regularization on the main effects, RAMP substantially cuts down on the size of the model matrix in its second stage, where it only considers candidate interactions that made it past the first stage of selection. All of these methods constrain the solution path to weakly or strongly hierarchical models.

One consideration that is often mentioned only briefly in these other works is whether or not we *should* restrict all candidate models to be hierarchical. Authors usually note that most of the time, the answer is yes. However, Chipman (1996) provides a compelling paradigm for model hierarchy in a Bayesian context, and especially why it may not always be a safe restriction. If we think of interactions as children of their "parent" main effects, we would guess that a child is certainly *most likely* to be in a model if its parents are both in the model. It is comparably less probable that a child is in a model if one of its parents is not. Is it

absolutely impossible (with probability zero) for a child to be in a model without either of its parents?

The answer is no; there are numerous occasions where a generating model is not hierarchical. Chipman gives the example of the atmospheric sciences, where relations of the form $Y = A\exp(BC)$ are common, which is a non-hierarchical model on the log scale. We can also point to models for lung cancer, where "pack years" is an acknowledged risk factor. In fact, a non-hierarchical model of this type is plausible in any setting where level of exposure and time of exposure are both captured somewhere in the candidate covariate space. Therefore, we argue that in lieu of hierarchy constraints, a better general rule would be to enforce hierarchy *preference*. This is considerably different than a *constraint*. In the next section, we will show that the SRL enforces higher penalties for interactions than for the main effects (when $p > 2$), which naturally enables hierarchy preference (but does not force hierarchy).

It is important to note that many of the methodological works on interaction selection involve a comparison to the *all-pairwise lasso* (APL); the lasso applied to the covariates and their interactions indiscriminately. In these comparisons, the APL consistently selects too many interactions. This issue gets compounded when the true generating model has very few "active" interactions, which is an ongoing limitation of interaction feature selection for some of these methods (Lim and Hastie, 2015).

In section 2.3, we compare the performance of SRL to the APL and to glinternet in selecting from all pairwise interactions in a simulation study. We measure the performance of these frameworks in three ways. First, we use the root-mean-squared error (RMSE) on newly generated data to compare the the predictive accuracy of the final models. Second, we use the false discovery rate (FDR), the expected number of Type I errors, and the expected number of Type II errors, examining these quantities both collectively and separately for interactions and main effects. Third, we test the computation time under several different generating model scenarios. A brief exploration of similar ranked-sparsity techniques for nonconvex regularization is presented in section 2.3.5. Finally, we compare more general strengths and weaknesses of the SRL relative to glinternet and other methods in the discussion.

2.2 Implementation of *Ranked Sparsity* via the Lasso

Suppose we have features $[x_1, x_2, ..., x_p] = X_{nxp}$, and a centered response variable $\boldsymbol{y}$, and we suspect that some (but not all) features are related to $\boldsymbol{y}$. The lasso solution can be obtained by minimizing the following expression with respect to $\boldsymbol{\beta}$:

$$||y - X\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1$$

There is a Bayesian motivation for this solution. If each $\beta_j \overset{iid}{\sim} \text{Laplace}(0, \lambda)$ *a priori* (which is another way of characterizing our prior suspicion that some, but not all, $\beta_j = 0$), then the *mode of the joint posterior distribution* represents the lasso solution (Tibshirani, 1996). This concept can be visualized in a shiny app that we have developed, which is hosted at https://ph-shiny.iowa.uiowa.edu/rpterson/shiny_vis1/. More information about this concept can be found in Strawderman, Wells and Schifano (2013).

While many authors have investigated a more fully Bayesian approach to the lasso problem, we will only use the Bayesian interpretation as a starting ground. For a likelihood-based lasso procedure, the prior on the coefficients can be written as

$$\pi(\boldsymbol{\beta}) = \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda|\beta_j|}$$

Typically, λ is thought to be "known" in the actual estimation of $\boldsymbol{\beta}$, and the "solution path" is formed by locating the posterior mode at a grid of λ values. The selection of λ has important predictive and interpretative consequences – higher values will yield more parsimonious models and will shrink coefficients to a greater extent. The selection of an optimal λ, whether via a Bayesian estimation procedure or cross-validation, is essential for interpretation (and, in the absence of model averaging, prediction as well). Selecting an optimal λ can also be thought of as a direct way to answer Einstein's version of Occam's Razor: one should increase λ as long as predictive accuracy does not suffer, but no more. Again, in this formulation, parsimony is equated to simplicity.

As a brief aside, we wish to invoke a seemingly unrelated topic: Fisher information. Fisher information is regularly utilized in likelihood theory to describe the behavior of maximum likelihood estimators, but at its heart, it is a measure that can quantify the structural characteristics of any joint density. Jeffreys derived

his famous "noninformative" prior based on the concept of the Fisher information of a *prior* density. For $W \sim f(w|\lambda)$, where $\lambda \in \Lambda$ is scalar and $\lambda \to \log f(w|\lambda)$ is twice differentiable in λ for every w, the model Fisher information at any λ is defined to be

$$I(\lambda) = E_{W|\lambda}\left[-\frac{\partial^2}{\partial\lambda^2}\log f(W|\lambda)\right]$$

Returning now to the problem at hand, consider partitioning X into K groups, such that

$$X = [A_1, A_2, ..., A_k, ..., A_K]$$

Let p_k refer to the column dimension of $A_k \; \forall \; k$, and let β_j^k refer to a particular β_j in covariate group k, then the prior for $\boldsymbol{\beta}$ undergoes a purely cosmetic change and becomes

$$\pi(\boldsymbol{\beta}|\lambda) \propto \prod_{k=1}^{K}\prod_{j=1}^{p_k}\lambda e^{-\lambda|\beta_j^k|}$$

Now, take λ to be a parameter. Since we are still formulating a prior, all of the β_j^k can be thought of as random variables. It then becomes straightforward to find the Fisher information in this prior density (see chapter 1 for this derivation):

$$I(\lambda) = E_{X|\lambda}\left[-\frac{\partial^2}{\partial\lambda^2}\log\pi(\boldsymbol{\beta}|\lambda)\right] = \frac{1}{\lambda^2}\sum_{k=1}^{K}p_k$$

The information increases with the dimension of each group's parameter space equally (for any $\lambda > 0$). In other words, if $p_1 = 10$, and $p_2 = 100$, by default the covariate group A_1 contributes a tenth of the prior information as does the covariate group A_2, for any $\lambda > 0$. If A_1 refers to the main effects, and A_2 refers to their pairwise interactions, it becomes clear why the APL typically performs terribly; the *a priori* informational asymmetry between the interactions and the main effects leads to too many Type I errors among the candidate interaction effects, and too much shrinkage among the main effects.

In many (perhaps most) situations, the preceding weighting scheme may not be desired. We can slightly modify the prior distribution by replacing λ with $\lambda_k = \lambda\sqrt{p_k}$. Now, unlike before when the distributions were independent and identical for all k, each β_j^k is only independent and identically distributed within its own covariate group k. Within a group, coefficients have Laplace prior distributions with mean 0 and variance $\frac{1}{\lambda^2 p_k}$.

The Fisher information contained in the prior for covariates in group k after this modification is

$$I(\lambda_k) = \frac{p_k}{\lambda_k^2} = \frac{p_k}{\lambda^2 p_k} = \frac{1}{\lambda^2} \; \forall \; k$$

In words, by scaling each group's penalty by the square-root of its dimension, we have ensured that the prior information is the same across groups; no group has an *a priori* informational advantage. If we add another tuning parameter, γ, in the definition for λ_k such that $\lambda_k = \lambda p_k^{\gamma}$, the resulting approach can be seen as a generalization the ordinary lasso, where

$$I(\lambda_k) = \frac{1}{\lambda^2} p_k^{(1-2\gamma)}.$$

If $\gamma = 0$, this is identical to the ordinary lasso. If $\gamma = 0.5$, each covariate group contributes the same amount of prior information (which is a good default setting for many circumstances). As γ increases, the penalties for larger groups of covariates increase quickly (as the information contribution decreases quickly with group size).

We consider a final variant of the penalty weights where the information is expected to decrease as the *group index* increases. Specifically, instead of weighting the penalties by $\sqrt{p_k}$, we use $\sqrt{\sum_{i=1}^{k} p_i}$. This yields a group-level information contribution of

$$I(\lambda_k) = \frac{1}{\lambda^2} \left(\frac{p_k}{\sum_{i=1}^{k} p_i} \right)$$

In words, the information contribution is highest for A_1, and decreases cumulatively as more groups are added. If A_1 represents main effects, and A_2 represents squared polynomial terms of those main effects, this cumulative group index penalty ensures that the polynomial terms are penalized more heavily than the main effects.

We call these modifications of the lasso the *sparsity-ranked lasso* (SRL), and we explore the performance of the SRL in the forthcoming simulations and application.

2.3 Simulations

2.3.1 Simple Simulation Study

Consider a simple simulated example, where we have 100 observations arising from a true $f(x) = 10 * (x - .5)^2$ measured with some residual noise, $\varepsilon_i \overset{iid}{\sim} N(0, 0.9^2)$, and $x \sim \text{unif}(0, 1)$. This relationship is shown by the solid black line in Figure 2.1. It is well-known that the addition of extraneous polynomial terms in a regression model hurts the model's performance, especially at the bounds of the covariate space. This can be seen in the top three plots of Figure 2.1 – adding higher order terms increases the "wiggliness" of the fits (represented by the grey lines).

With only one covariate, we could easily fit several models with increasing orders of polynomials and select the best using an information criterion. However, in higher dimensional settings, this approach is not a practical option. Another option that is more feasible in high-dimensional settings is to use the lasso to select the optimal order interaction – this method is explored in the middle 3 plots of Figure 2.1. Evidently, the bias incurred by the L1 penalty reduces some of the variability (the "wiggliness") in the relationship, while at the same time contaminating the shape of relationship (note that the fitted lines are bent down towards the origin). Note that in these simulations the optimal λ is selected with the Bayesian Information Criterion (BIC).

We can also investigate what happens when we use the SRL in this simple setting. In the SRL setup, $\lambda_j = \lambda(d_j)^\gamma$, where d_j refers to the degree of covariate j. This approach is equivalent to cumulative group penalties in the 1-covariate case. The resulting fits are shown by the grey lines in the bottom three plots of Figure 2.1. We observe that the fits are both reducing the wiggliness from the extraneous terms while at the same time inducing less "bending" towards the origin. Again, the optimal λ and γ are selected with BIC.

The plots in Figure 2.1 only show 50 fits each, but repeating this process 10,000 times, we can compare the mean squared error (MSE) of each method across the domain of x. In Figure 2.2, we show the increase in the MSE for each method relative to a baseline "oracle" model (i.e. an OLS model that only includes the "true" x and x^2).

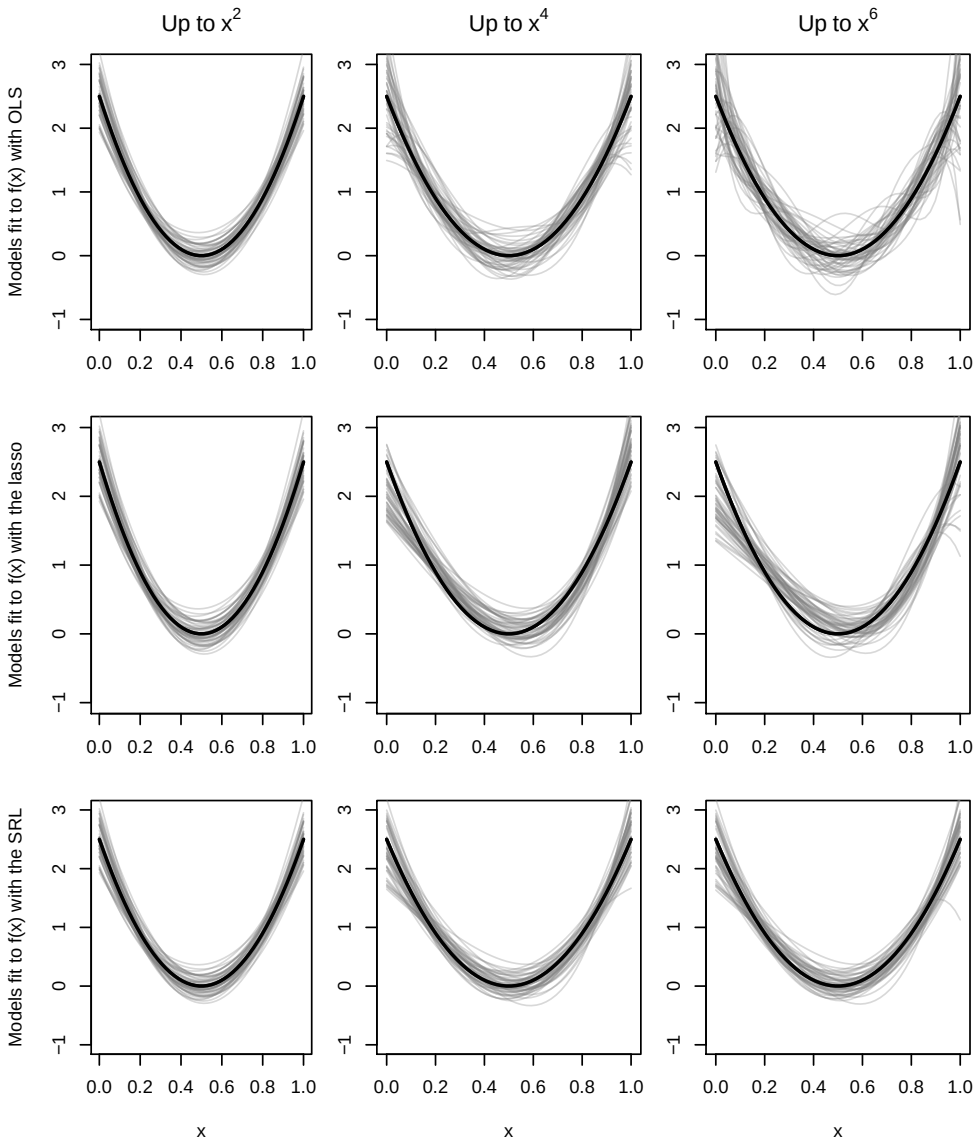

Figure 2.1: A simple simulation where 50 samples of size 100 are generated for x and a response variable y with the relationship $y = f(x) + N(0, .9^2)$. The black line represents the true f, and the grey lines represent 50 fits to the different samples. Models in the top three plots are fit using ordinary least squares (OLS); in the middle three plots, models are fit with the lasso; and on the bottom three plots, models are fit using the sparsity-ranked lasso (SRL). The covariates included are the polynomials of x up to x^2 (left) up to x^4 (center), and up to x^6 (right).

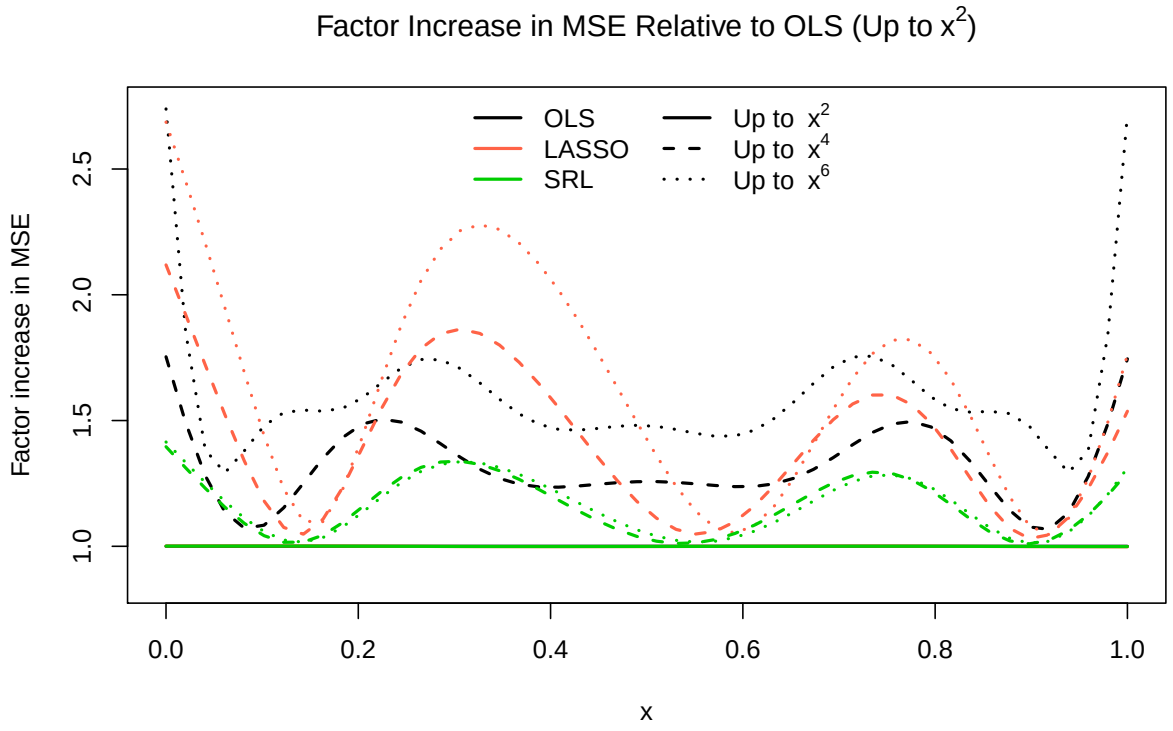

Figure 2.2: The expected increase in the mean squared error (MSE) for the ordinary least squares (OLS), lasso, and sparsity-ranked lasso (SRL) models relative to a baseline "oracle" OLS model that only includes the "true" variables x and x^2).

Figure 2.2 shows that while there is no replacement for an "oracle" model, the next best models are those which utilize the SRL method. Interestingly, there appears to be very little in terms of predictive difference between the SRL applied up to the 4th order and SRL applied up to the 6th order; this implies that we could likely increase the order and still not observe a substantive impact on the predictive performance. On the other hand, if the lasso is used, there is a marked decrease in predictive performance between the 4th order model and the 6th order model; it appears as though these models (as well as the OLS models) perform increasingly poorly as the number of extraneous polynomials increases. Since this is a very specific setting, the next section generalizes this simulation to a larger class of polynomial generating models.

SRL vs. other smoothers

As we have seen in section 2.1.3, there are many options for smoothers in this low-dimensional setting. Here, we explore to some extent how well the SRL performs relative to these other methods in another simulation. We investigate four possible generating models: polynomials of orders 10, 2 (quadratic), 1 (linear),

and 0 (null). In each setting, we generate data ($n = 100$) according to the following model:

$$y = \alpha + \beta_1 x + \beta_2 x^2 + ... + \beta_{10} x^{10} + N(0, 1)$$

For the high-order setting, β_j parameters were generated randomly. First, we draw $\theta_1, \theta_2, ..., \theta_{10} \sim N(0, 1)$, then we scale them so their magnitude sums to 10; $\beta_j = 10 * \theta_j / \sum |\theta_i|$. The same technique was used in the quadratic generating model, except only for $j \in \{1, 2\}$; all other parameters were set to 0. In the linear case, $\beta_1 = 10$, and in the null setting, $\beta_j = 0 \ \forall j$. The sole covariate x follows a standard uniform distribution within each simulation.

For model fitting, we utilize the SRL method with cumulative weights on the degree of the polynomial up to the 10th order. We compare this model fit with the LOESS smoother (`loess()` in the `stats` package) and with a smoothing spline (the `gam()` and `s()` functions from the `mgcv` package). The default tuning settings are used for these functions. SRL is tuned using repeated ($r = 5$) 10-fold cross-validation with $\gamma \in \{0, .5, 1\}$. The simulations are repeated 1,500 times. Models are evaluated on the basis of the root-mean-squared error (RMSE) relative to the expected value of $Y|x$ on $n = 10,000$ new observations. We use the following definition of RMSE for this simulation study and throughout the dissertation:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - E(y_i|x_i))}$$

The results of this simulation presented in Figure 2.3 show that the SRL method performs quite well compared to LOESS and spline alternatives; even slightly better. This is especially the case in the lower order and null models, where the SRL is performing almost as well as the "oracle" model in terms of prediction. In the null setting, the smoothers are tending to overfit more than the SRL method; SRL improves upon the other methods in this null setting. This is important to recognize, as it will come into play in the high-dimensional setting where we expect many null relationships. For the 10-degree generating model, the poor performance of the "full" model, despite it being technically correctly specified, is a mark of the fact that there is high correlation among the polynomials of x, which inflates the variance of these coefficients.

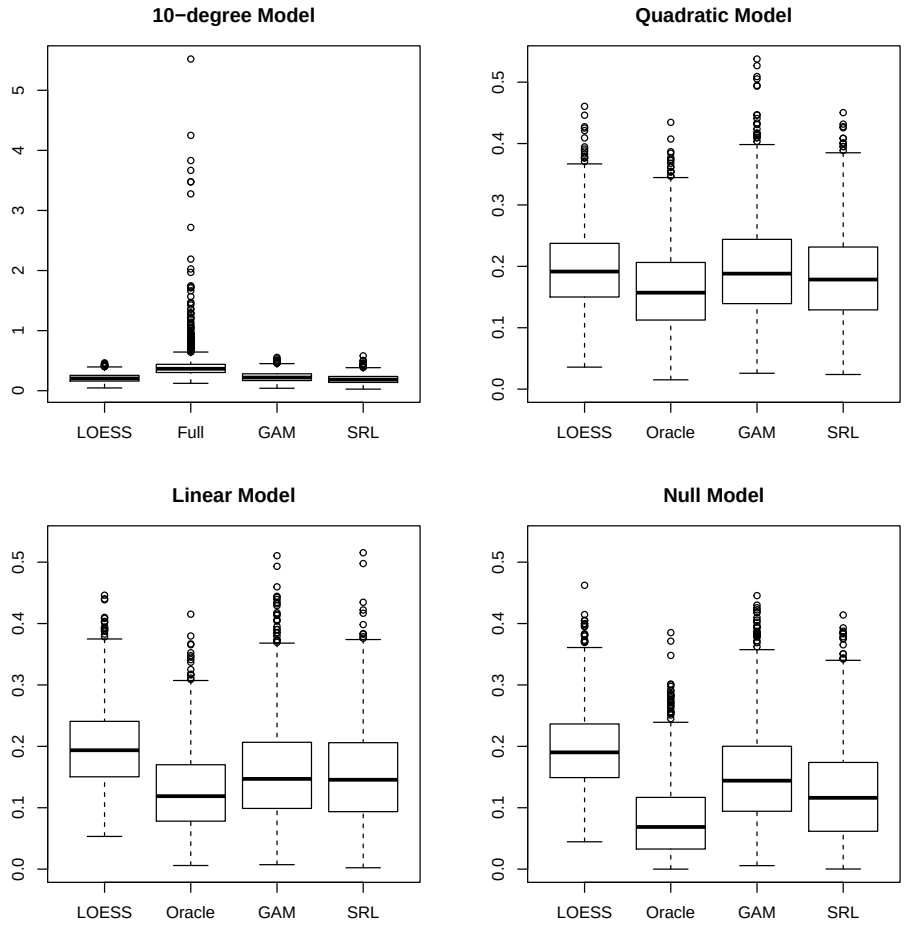

Figure 2.3: Performance of smoothing methods in describing a truly polynomial (or null) relationship between a single covariate and response. The RMSE within each simulation is plotted along the y-axis.

Admittedly, this simulation is biased in favor of SRL methods because the generative setting is truly a polynomial. Also, the covariate has no skew, so the higher order polynomials will not have highly influential points. We have simulated what happens under skewed distributions, and we found that the SRL techniques generally begin to perform worse relative the other smoothers as the skew increases. However, a normalizing transformation on the covariate prior to expanding the polynomial can mitigate this effect. We have developed software in a separate work that can adequately and robustly perform these normalizations (Peterson, 2017; Peterson and Cavanaugh, 2019). For non-polynomial generating distributions, the other smoothing techniques typically outperform the SRL and other traditional polynomial-based methods.

2.3.2 Extended Simulation Study

2.3.2.1 Simulation Setup – Polynomials

While we have shown how the SRL can compete with other smoothing techniques in the 1-dimensional setting, the true benefits of the SRL methodology are present in the medium-to-high dimensional setting where model selection must take place. In these settings, if one attempts to fit univariate splines to each of the covariates, we have seen that the splines (or kernel-based methods) tend to overfit the truly "null" relationships (as well as the truly "linear" relationships). Further, splines and kernel-based methods are more challenging to implement computationally. Splines in particular become intractable mathematically if the lasso is used to simultaneously select and estimate the parameters in the model, since the lasso shrinks each coefficient to zero. Therefore in this setting, the SRL polynomial smoothing method can truly shine. In this section, we show how the SRL compares to the LASSO in selecting from truly polynomial relationships in a sparse, medium-dimensional setting.

Let $\mathbb{X} = \left[X, X^{\odot 2}\right]$ refer to the column combination of the main covariates (an $n \times p$ matrix), followed by their element-wise squared values (also $n \times p$). We wish to fit the following linear model:

$$\boldsymbol{y} = \mathbb{X}\boldsymbol{\beta}$$

We partition $\boldsymbol{\beta}^T = \left[\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T\right]$ to correspond with our notation for $\mathbb{X}$. In "the usual case", where polynomials are not considered, it is assumed that $\boldsymbol{\beta}_1$ is the only parameter vector with nonzero components. One would expect that this assumption helps in situations where the true generating model is, in fact, linear. However, what if there are nonzero components in the other parameter vector? We will investigate. In the simulations to follow, we take $n = 200, p = 20$, and we generate each element in X as independent uniform(0,1) random variables. In addition, we set the number of nonzero main effects as $s = 5$. In order to generate our $\boldsymbol{\beta}$ coefficients in such a way that a large set of possible f relationships are considered, we use scaled normal random variables as our "active" (i.e. nonzero) parameters in $\boldsymbol{\beta}$, as described in the next paragraph.

We investigate six different generative settings with different numbers of active polynomials $b \in \{0, 1, 2, 3, 4, 5\}$. Given b, we begin a simulation by drawing $\theta_1, \theta_2, ..., \theta_s \sim N(0, 1)$. Then, we compute the active linear parameters: $\beta_{1j} = 10 * \theta_j / \sum |\theta_i|$. In the settings where there are active polynomial effects

(i.e. $b > 0$), we then generate b standard normal variables ϕ, and scale them to compute the polynomial effects $\beta_{2j} = 10 * \sqrt{15}\phi_j / \sum |\phi_i|$. The addition of the $\sqrt{15}$ is necessary to compensate for the difference in the variance between a covariate and its square. This algorithm of generating the $\boldsymbol{\beta}_j$ parameters ensures that for $b \neq 0$, a wide variety of true polynomial relationships are considered, while keeping the overall amount of signal the same across simulations.

In order to fit these models, we consider five modeling frameworks: LS0 (lasso on original terms only), LS1 (lasso with original and squared terms), LS2 (lasso with up to 3rd order polynomials), SRL1 (sparsity-ranked lasso with original and squared terms), and SRL2 (sparsity-ranked lasso with up to 3rd order polynomials). Models are tuned with BIC, then used to predict 10,000 new observations. This process (including the new generation of β_{ij} terms) is repeated 10,000 times, after which we compare the ratio of the RMSE of the predictions in the lasso models divided by the RMSE in the SRL models.

The predictive performance is shown in Figure 2.4. The lines representing the performance of the lasso models have different SRL baselines; LS0 and LS1 are compared to SRL1, and LS2 is compared to SRL2. We see that LS0 exhibits a large loss in predictive performance when there is one or more active squared terms. However, if the true model does not have a squared term, LS0's presumption that $\boldsymbol{\beta}_2 = \mathbf{0}$ does provide a minuscule improvement in performance. The SRL1 method performs considerably better than the LS1 model for especially sparse polynomial models. This improvement becomes more pronounced when looking at SRL2 compared to LS2; since there are no true 3rd order polynomial effects, LS2 is falling into the trap of shrinking the first and second order terms too much while selecting too many 3rd order terms. Note that for figures in this chapter, the "^" notation refers to the values of the LS0 model that were too large to be clearly plotted next to the other curves.

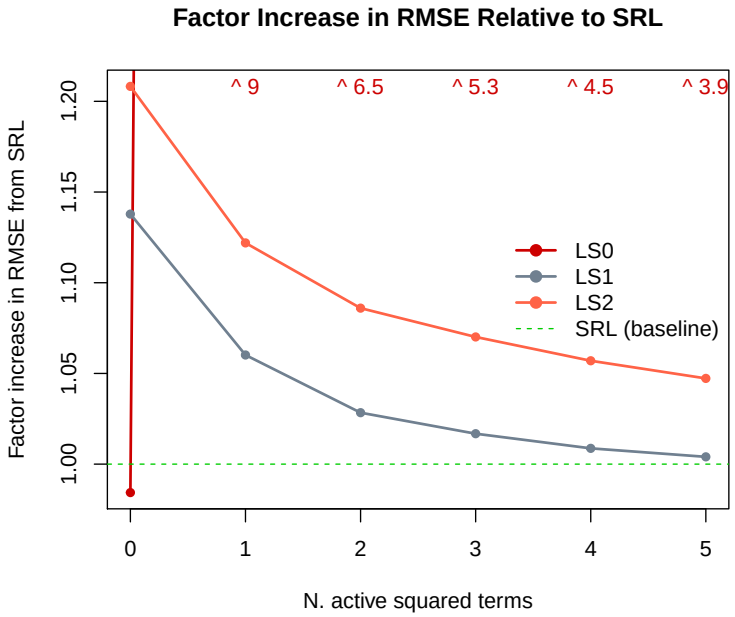

Figure 2.4: Predictive performance of various polynomial fitting methods relative to the SRL. LS0 refers to the lasso fit using only the original terms; the line is relative to SRL1. LS1 and SRL1 refer to the lasso fit and the sparsity-ranked lasso fit (respectively); these models include original and squared terms. The line for LS1 is relative to SRL1. LS2 and SRL2 refer to the lasso fit and the sparsity-ranked lasso fit (respectively); these models include original, squared, and cubed terms. The line for LS2 is relative to SRL2. The "^" notation refers to the values of the LS0 model that were too large to be clearly plotted next to the other curves.

The plots in Figure 2.5 show model selection information for each modeling framework. We can see that the SRL methods perform very well in the setting when there are no true polynomial terms; the FDR and the number of Type I/II errors is very similar to the LS0 model and lower than the LS1/LS2 models. This is due to the SRL having a much lower FDR and Type I error rate in the polynomial terms than the LS methods. While the number of Type I errors is generally increasing with the number of active polynomials for SRL, this is compensating for the fact that LS0 is failing to select active polynomials (as evidenced by the large Type II error rate for LS0; it is missing all of the active polynomials by default). Finally, note that the SRL1 and SRL2 methods perform similarly, whereas LS1 and LS2 perform very differently (LS2 performs much worse). Therefore, these simulations provide some evidence that the SRL method can effectively select from a set of possibly extraneous polynomial terms in a high-dimensional setting.

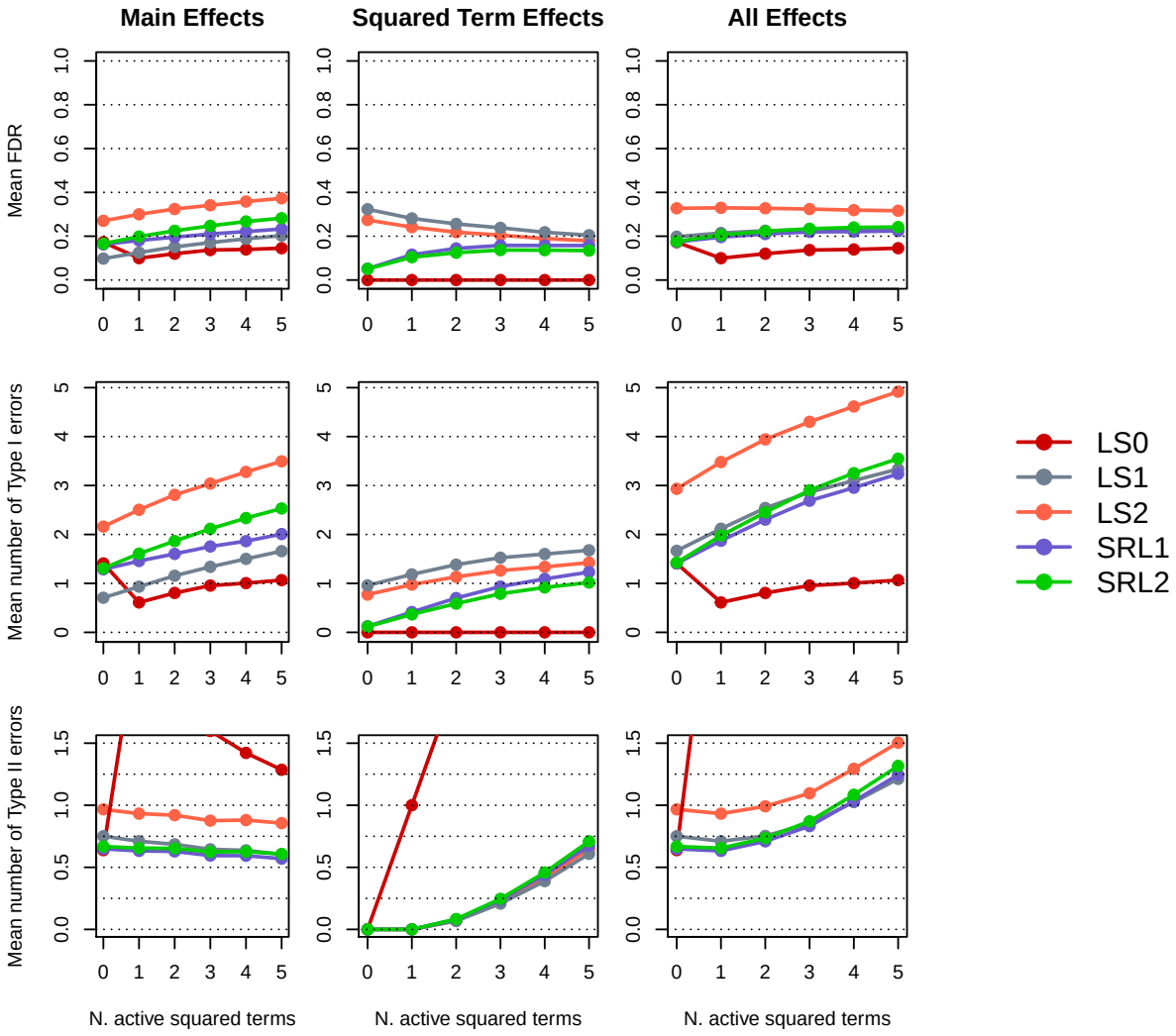

Figure 2.5: Model selection performance of various polynomial fitting methods. The top three plots show the mean FDR across simulations, the middle three plots show the mean number of Type I errors, and the bottom three plots show the mean number of Type II errors. The metrics are stratified into main-effects (left), squared effects (center), and their combined/overall values which also potentially includes cubed effects (right). LS: lasso, SRL: sparsity-ranked lasso. The 1 postfix indicates main effects and squared terms were included, and the 2 postfix indicates that polynomials up to order 3 were considered. LS0 refers to the lasso with no polynomials considered.

2.3.2.2 Simulation Setup – Interactions

We set up a similar simulation in the context of interactions; now instead of active squared terms, we have active pairwise interactions. In this context we also consider the glinternet method as a benchmark in

addition to the all-pairwise lasso (APL) and LS0 (the lasso with only the original covariates).

In this simulation, we take $n = 300$, and $p = 20$. Since the glinternet method uses 10-fold cross-validation to tune λ, we do likewise for SRL instead of BIC to ensure a fair comparison (this will increase the FDR and the number of Type I errors while improving predictive performance). The γ parameter for SRL is fixed to 0.5, corresponding to an equal contribution of prior information from the main effects and the interaction effects. The algorithm to generate the nonzero (active) coefficients is similar to the polynomial simulation, where we have 11 generative settings of interest corresponding to the number of active interactions b. The algorithm is comprised of the following steps.

- For $b \in \{0, 1, 2, ..., 10\}$,
  - Draw $\theta_1, \theta_2, ..., \theta_s \sim N(0, 1)$
  - Compute $\beta_{1j} = 10 * \theta_j / \sum |\theta_i|$
  - Generate b standard normal variables ϕ
  - Select the index of active interactions j according the rules outlined in the following paragraph
  - Set $\beta_{2j} = 10 * \sqrt{\frac{7}{12}} \phi_j / \sum |\phi_j|$

The generating models were not necessarily strongly hierarchical. In particular, each simulation was set up according to Chipman's paradigm; strong hierarchy was most probable, weak hierarchy less so, and anti-hierarchy least so. Given $s = 5$ and $p = 20$, there are $\binom{s}{2} = 10$ candidate interactions that would yield a strongly hierarchical model, $s(p - s) = 75$ that would yield a weakly hierarchical model, and $\binom{p-s}{2} = 105$ that would yield an anti-hierarchical model. Within a simulation, each active interaction effect was drawn at random from these bins of "strong", "weak" or "anti" candidate effects with probabilities 0.7, 0.2, 0.1, respectively.

In order to fit these models, we consider four modeling frameworks: LS0 (lasso on original terms only), APL (lasso with original and all pairwise interaction terms), SRL (sparsity-ranked lasso with original and all pairwise interaction terms), and GLN (glinternet model). For each framework, the optimal λ is selected with 10-fold CV, then that tuned model is used to predict 10,000 new observations. This process (including the new generation of β_{ij} terms) is repeated 1,000 times, after which we compare the ratio of the RMSE in each model relative to the SRL model.

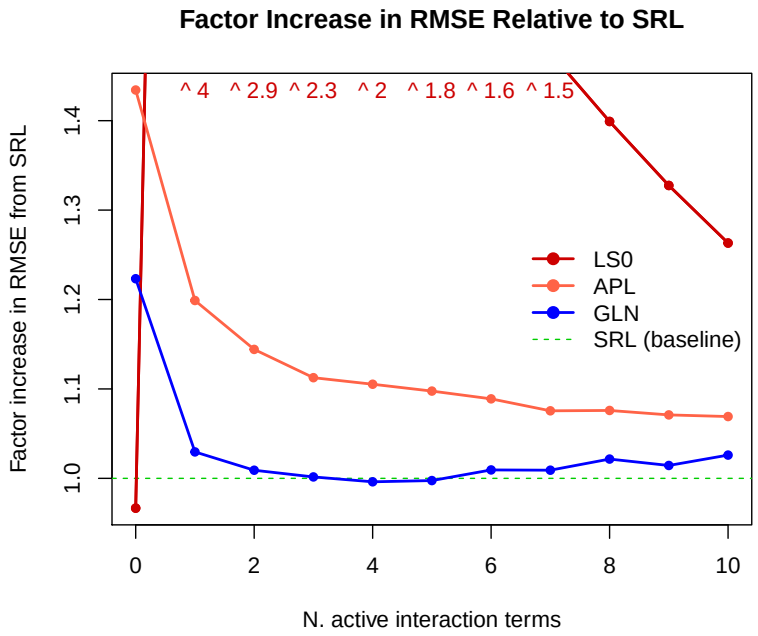

Figure 2.6: Predictive performance of various interaction fitting methods relative to SRL. LS0 refers to the lasso fit using only the original terms, APL refers to the lasso fit using the original terms and all pairwise interactions, SRL refers to the sparsity-ranked lasso fit with $\gamma = 0.5$, and GLN refers to the glinternet model. For all models, λ was tuned with 10-fold cross-validation. The "$\widehat{\ }$" notation refers to the values of the LS0 model that were too large to be clearly plotted next to the other curves.

The predictive performance of the models is shown in Figure 2.6. As was true in the polynomial simulation, the LS0 model exhibited a very large loss in performance when an active interaction was present (and demonstrated a very slight gain in predictive performance if the true model had no interactions). The APL model performed comparably better than the LS0 model when any interactions were present, but performed much worse in the no-interaction case. If there were any active interactions, SRL performed much better than either the APL or the LS0. If there were no active interactions, SRL performed much better than either the APL or glinternet. SRL and glinternet performed similarly to each other when active interactions were present; this comparison is further outlined in the discussion section of this chapter.

The plots in Figure 2.7 show model selection information for each framework. When the true model has no active interactions, the LS0 and the SRL methods look very similar in terms of FDR and the mean number of type I/II errors. In this same setting, the glinternet and APL models have a much higher FDR and mean number Type I errors; this is driven by the tendency of these models to select too many interactions

(all of which are noise). For all of the generative settings, the overall FDR for the APL is very high, and it is driven disproportionately by a high FDR in the interaction effects. The glinternet method also exhibited this differential expression of the FDR among main and interaction effects, though to a lesser degree. This difference is further seen in the number of Type I errors; the higher FDR in the interaction effects translates to many more Type I errors for glinternet and APL than for SRL. The SRL method maintained approximately the same number of Type I errors in the interaction effects and the main effects for $b \geq 4$. This improvement in FDR/Type I error rate exhibited by SRL compared to glinternet is balanced out by a slightly higher mean number of Type II errors, a disparity which grows with the number of active interactions.

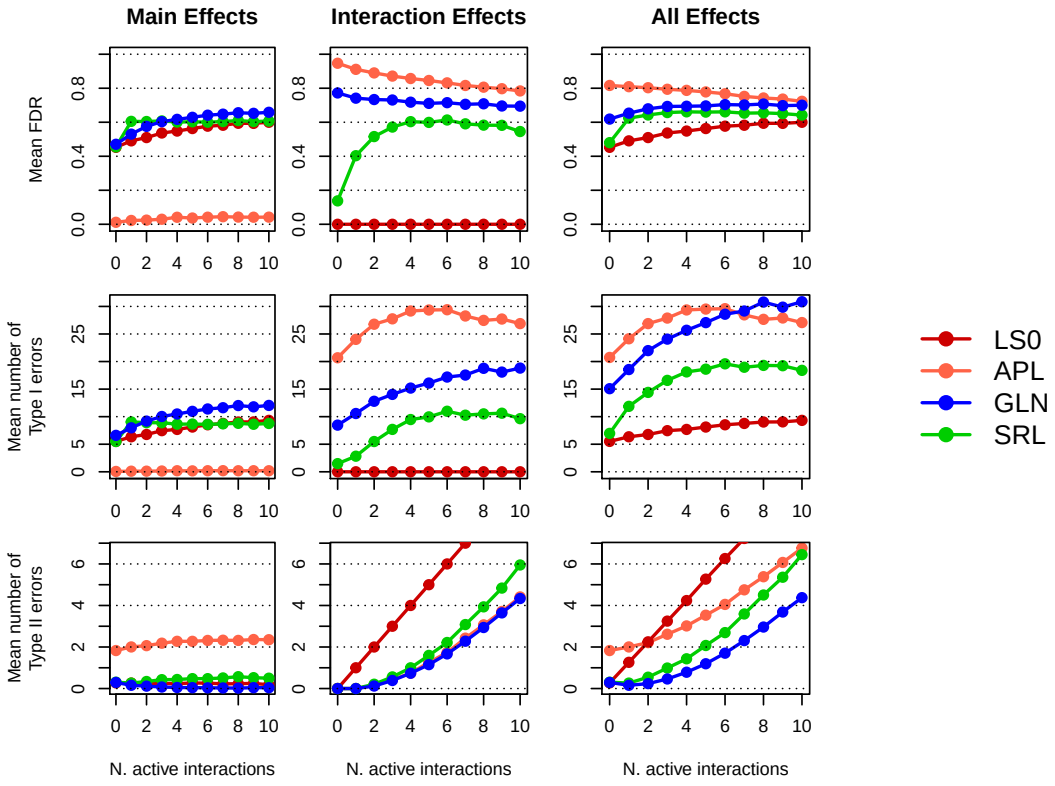

Figure 2.7: Model selection performance of various interaction fitting methods. The top three plots show the mean FDR across simulations, the middle three plots show the mean number of Type I errors, and the bottom three plots show the mean number of Type II errors. The metrics are stratified into main-effects (left), interaction effects (center), and their combined/overall values (right). LS0 refers to the lasso fit using only the original terms, APL refers to the lasso fit using the original terms and all pairwise interactions, SRL refers to the sparsity-ranked lasso fit with $\gamma = 0.5$, and GLN refers to the glinternet model. For all models, λ was tuned with 10-fold cross-validation.

2.3.3 Timing Considerations

To examine the difference in timing results between our method and `glinternet`, we set up a final simulation where $p = 25$ and the sample size varied from 25 to 10,000. The results presented in Figure 2.8 show the ratios of the mean run times for each method across 100 runs. The plotted values can be interpreted as the expected increase in the computational time relative to a model that does not consider any pairwise interactions. The SRL and APL methods perform close to one another, taking 2-6x more time to compute than LS0. On the other hand, glinternet takes between 40-100x more time to compute than LS0. This slow-down has more impact when cross-validation procedures are employed, as is the default means of tuning the λ parameter for glinternet. However, the other computational consideration of forming and storing the model matrix was not considered here, and could make the glinternet method more computationally feasible than the SRL or APL in some circumstances.

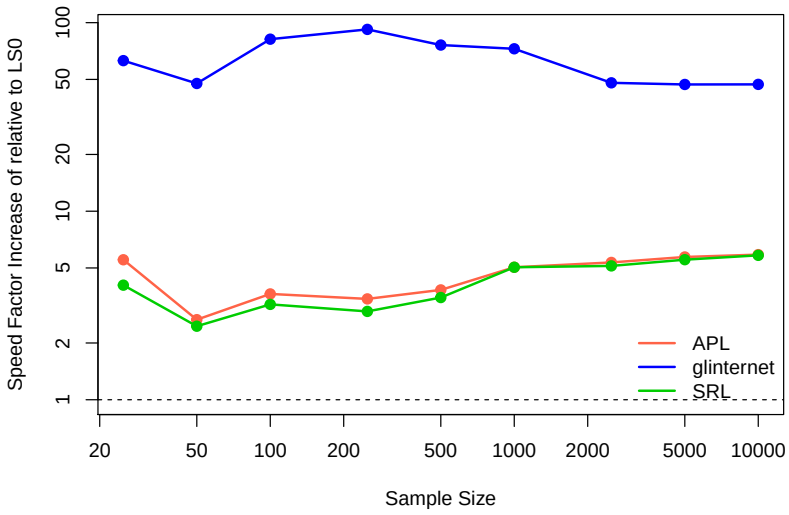

Figure 2.8: Expected factor increase in run time needed to consider all pairwise interactions among 25 predictors, relative to the run time on the main effects only (mean of 100 simulation runs). APL refers to the all-pairwise lasso, SRL refers to the sparsity-ranked lasso with $\gamma = .5$.

2.3.4 Nonconvex Regularization

In some settings, there are alternative regularization frameworks that can outperform the lasso. Two commonly used techniques that can reduce the bias incurred by the lasso are the Minimax Concave Penalty (MCP) (Zhang, 2010) and the Smoothly Clipped Absolute Deviations (SCAD) penalty (Fan and Li, 2001). Both of these methods "taper-off" the lasso's penalty for larger values of the coefficients, thereby reducing the bias for higher coefficients that gets induced by the lasso penalty; see section 1.4 for more information.

Unfortunately, there is no intuitive Bayesian interpretation of SCAD. Though there is a motivation for MCP in the Bayesian framework (Strawderman et al., 2013), the formulation is relatively complex, and without further investigation, it is unclear whether the same "prior information" logic that motivated the SRL holds for these other two settings. However, roughly speaking, the penalty parameter λ in either of these settings refers to the same concept of a "prior degree of skepticism". As λ decreases, more coefficients typically enter the model while non-zero estimates exhibit lessening shrinkage. It is possible to scale these penalties using penalty weights in the setting of MCP or SCAD, just as we did in the lasso setting. In this section, we implement a sparsity-ranked version of MCP and SCAD, which we abbreviate as SRM and SRS, respectively. We explore performance using the same extended simulation as in the previous section, investigating how these sparsity-ranked versions behave in the selection of interactions and polynomials relative to their non-sparsity-ranked counterparts (APM, APS for "all-pairwise" interactions, and SRMk, SRSk for degree k polynomial regression). A comparison of the predictive results for the polynomial and interaction settings are presented in Figures 2.9 and 2.10, respectively. Model selection performance in the interaction setting for MCP and SCAD, respectively, is presented in Figures 2.11 and 2.12.

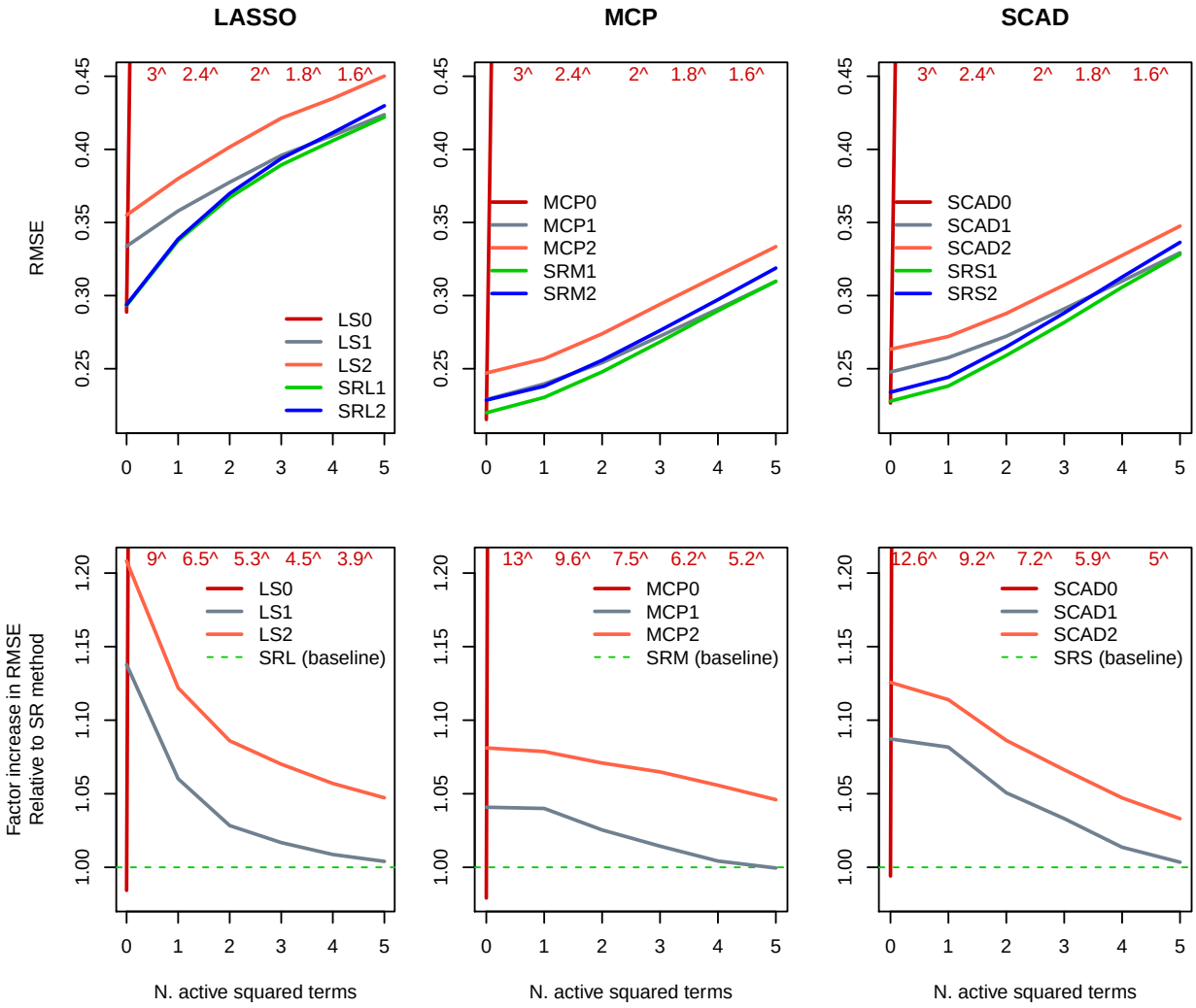

Figure 2.9: Predictive performance of sparsity-ranked nonconvex regularization in the polynomial setting. Absolute RMSE is plotted on top, and relative performance (to the corresponding sparsity-ranked method) is plotted on the bottom. The "^" notation refers to the values of the LS0 model that were too large to be clearly plotted next to the other curves.

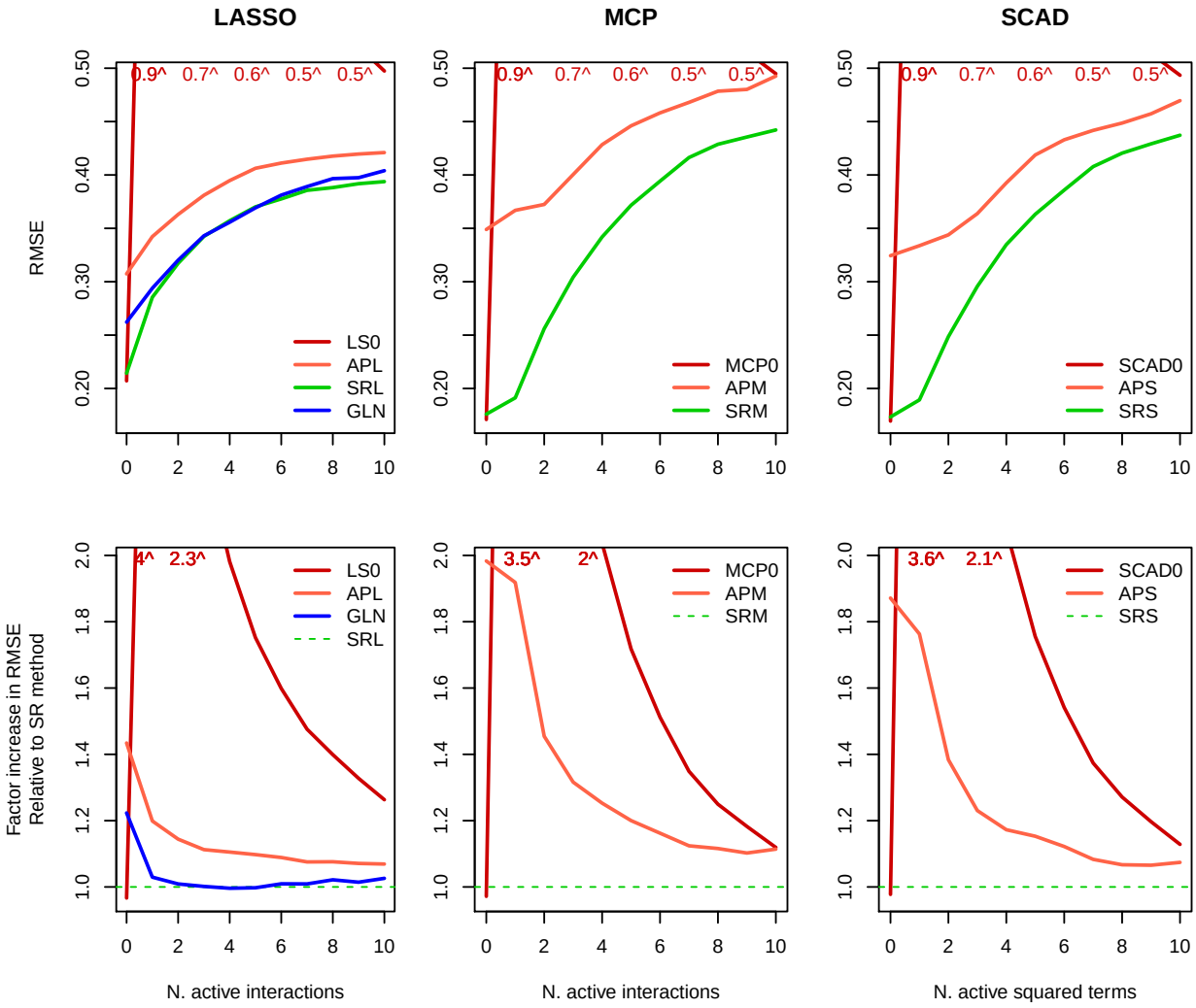

Figure 2.10: Predictive performance of sparsity-ranked nonconvex regularization in the interaction setting. Absolute RMSE is plotted on top, and relative performance (to the corresponding sparsity-ranked method) is plotted on the bottom. The "^" notation refers to the values of the LS0 model that were too large to be clearly plotted next to the other curves.

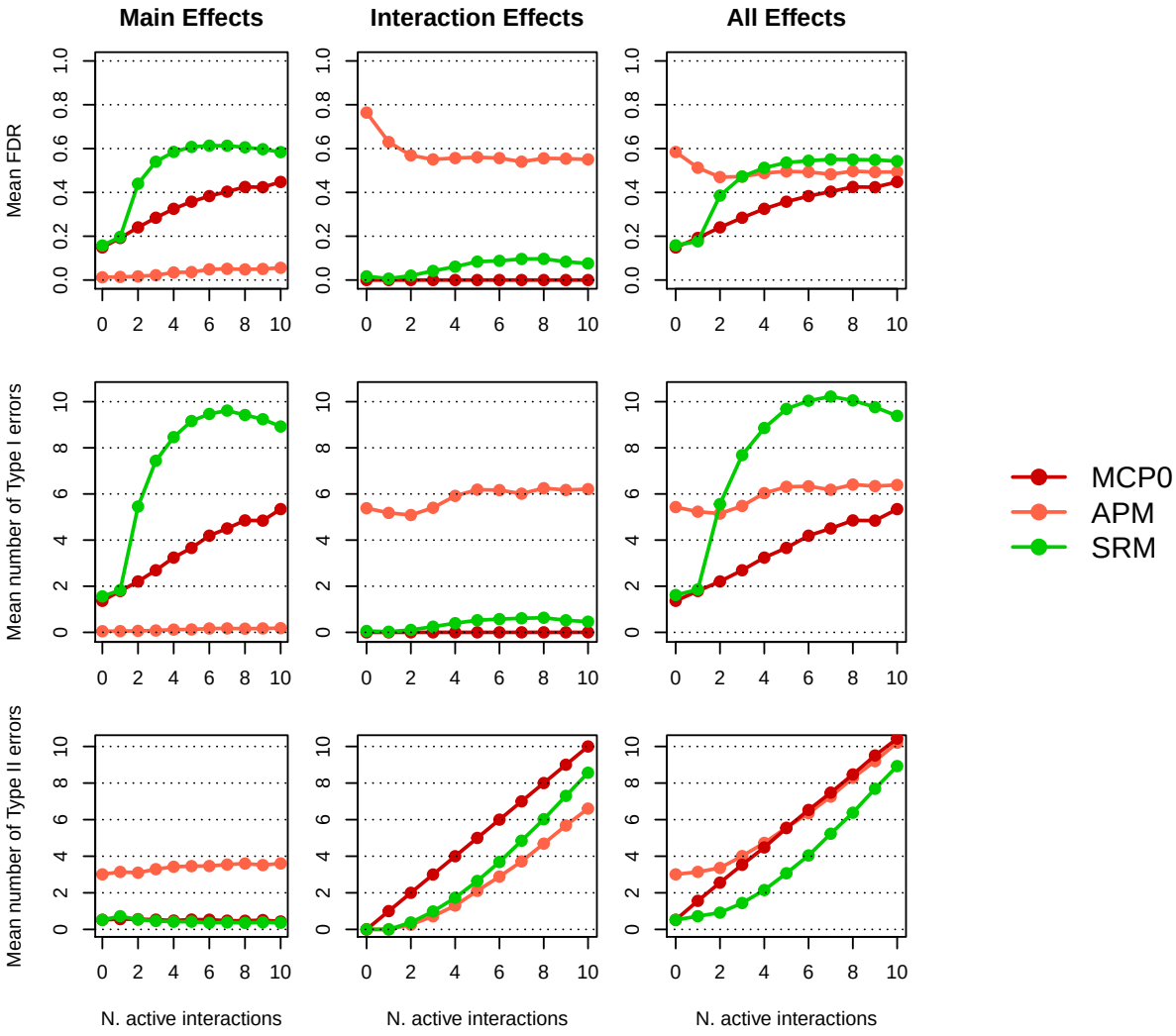

Figure 2.11: MCP model selection performance.

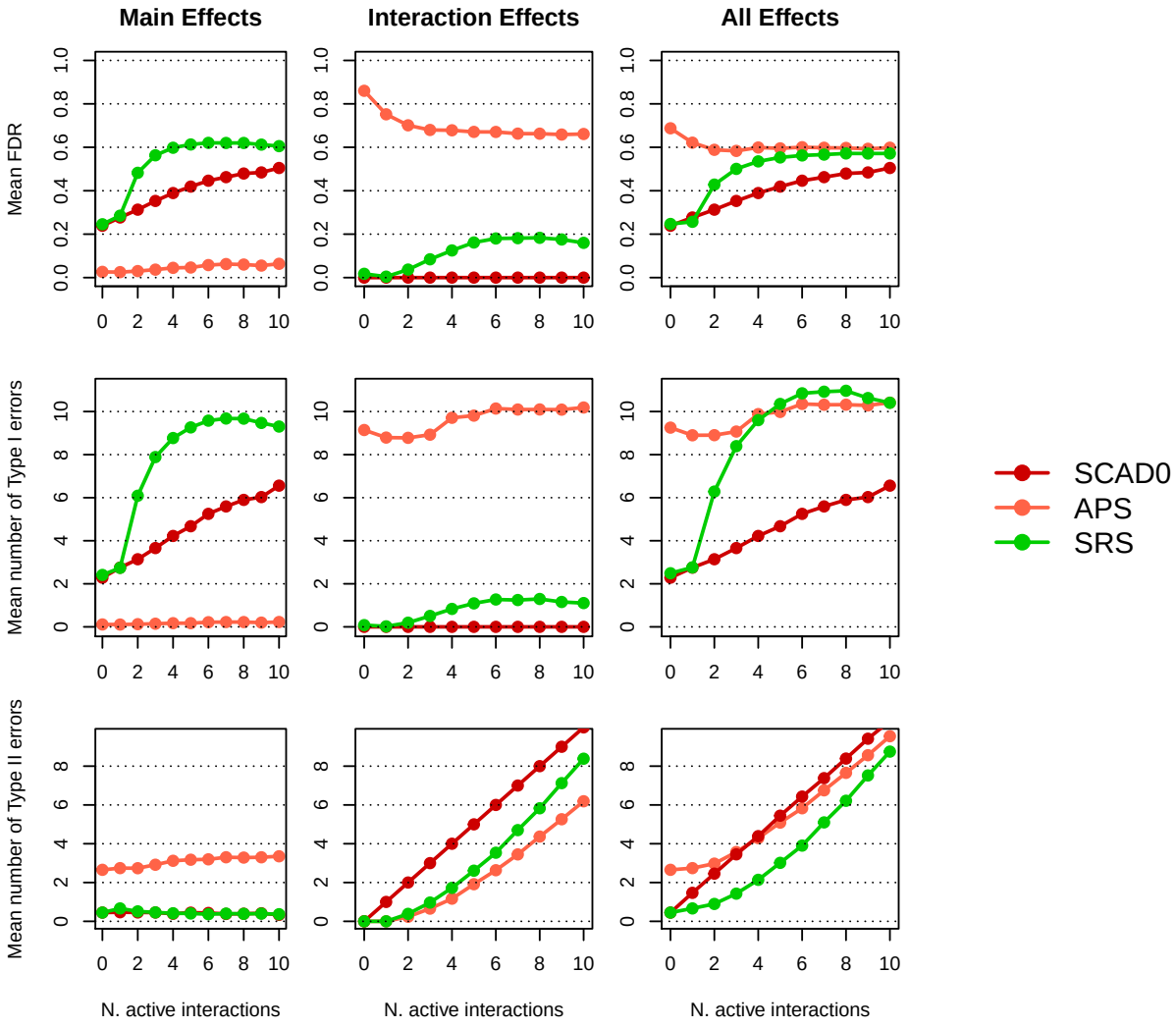

Figure 2.12: SCAD model selection performance.

In the polynomial setting (Figure 2.9), we see in the top plots that MCP and SCAD were able to predict substantially better than the lasso no matter how many polynomials were active in the generating model. In the bottom plots, we still see a very similar pattern for SRM and SRS that we observed for SRL; sparsity-ranked methods performed relatively better than their non-SR counterparts (though the difference is less pronounced in the setting with no active polynomials). In the interaction simulation, we see in Figure 2.10 that the "all-pairwise" MCP and SCAD (APM and APS) performed much worse than even the APL. SRM and SRS, however, were able to improve upon the best-case SRL model when the number of active interactions was low. In the bottom plots, we see that the relative performance of the SRM and SRS compared

to the APM and APS was very noticable; much more so than in the lasso case.

In Figures 2.11 and 2.12, we see quite similar model selection results for the SRM and the SRS, but the results look quite different than what we saw in Figure 2.7 for the sparsity-ranked lasso. Most notably, the false discovery rate among the main effects, along with the corresponding number of Type I errors, shoots up after the number of active interactions reaches 2. This does not occur in the interaction effects, where the FDR and the number of Type I errors stays quite low no matter how many interactions are active. With such a low FDR among interactions, one may wonder whether the SRM or SRS methods are selecting any interactions at all, but the bottom 3 plots indicate that they are; the number of Type II errors is less than the number of active interactions. For APM and APS, most of the false discoveries (Type I errors) were made in the interaction effects. This is a huge problem for the interpretation of these models; not only are they predicting worse, they are substantially harder to interpret due to the selection of many (truly spurious) interactions.

In summary, these results suggest that it is very important to account for ranked sparsity in the search for interactions even when using nonconvex regularization, and that simply scaling the penalty by the square-root of the dimension of the candidate covariate space is quite effective for this purpose.

2.4 Application: Gene-Environment Interactions

2.4.1 Background

We wish to show how SRL methods can be used in the context of genetic data, specifically for the purpose of detecting important gene-environment interactions. Gene-environment interactions make sense biologically, but unfortunately, they are very difficult to detect in practice. It is hard enough to detect an association with high-dimensional genetic data by itself, and looking for interactions with high-dimensional data is akin to searching for a few needles in tens of thousands of haystacks. We utilize a study that collected data on 442 patients with lung cancer (adenocarcinoma) (Shedden et al., 2008). For each patient, the investigators observed the time of death or censor (the primary outcome), 22,283 gene expression measurements taken from a sample of the lung cancer tumor, and some clinical covariates: sex, race, age, whether the patient received chemotherapy, smoking history, and cancer stage. The main outcome of overall

survival is presented in Figure 2.13.

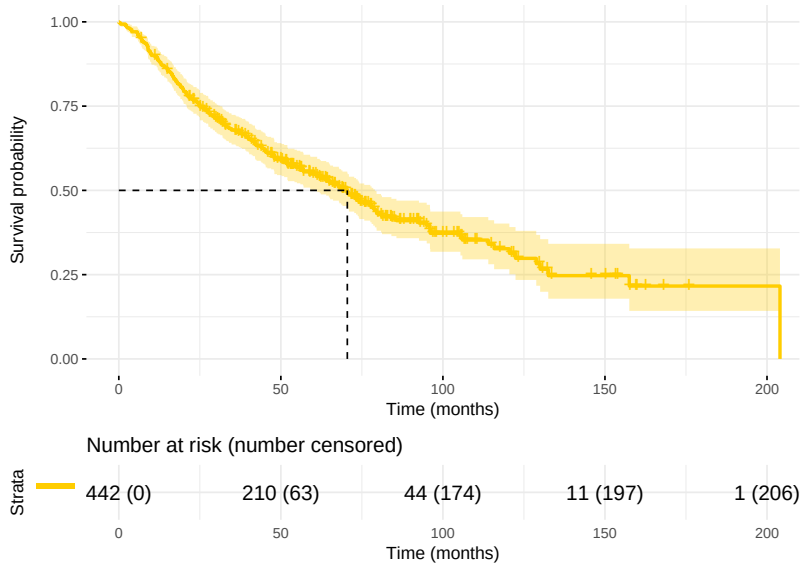

Figure 2.13: Kaplan-Meier curve for Sheddon data; survival time is the primary outcome of the study for which we build regularized Cox regression models to predict.

2.4.2 Methods

In order to fit gene-environment interaction models, we take our outcome $(\boldsymbol{y}, \boldsymbol{d})$ as the time until death/censor and the death indicator, respectively. We model the hazard of death using a Cox proportional hazards model with our predictors being comprised of the genetic and clinical covariates described in the previous section denoted by X and Z, respectively. For our purposes, we are only interested in interactions that may occur between X and Z or within Z; we do not look for interactions that occur within X (though this would be a straightforward extension).

We investigate the performance of 3 candidate modeling frameworks: the lasso using only main effects (LS), the sparsity-ranked lasso (SRL), and the all-pairwise lasso (APL). Within the SRL framework, we set $\gamma = .5$, and investigate three different penalty schema. SR1 refers to the SRL on the main and interaction effects with proportional weighting, and SR2 refers to the SRL with cumulative group-index weighting on the main and interaction effects. Since our features consist of both clinical and genetic covariates, we can also use SRL methods on the main effects by themselves in the original context of covariate groups of different

sizes; this is referred to as SR0. Unfortunately, we were not able to compare the SRL method in this context to another interaction-selection tool; to our knowledge, none of the regularization-based methods described in the introduction currently have open-source software available that can handle survival outcomes.

The first step in the modeling process is to split the data $\mathbb{X} = [\boldsymbol{y}, \boldsymbol{d}, X, Z]$ randomly into a training set $\mathbb{X}_{\text{train}}$ ($n = 342$) and a test set $\mathbb{X}_{\text{test}}$ ($n = 100$). Second, using $\mathbb{X}_{\text{train}}$, we use repeated ($r = 5$) cross-validation ($k = 10$) to tune each of the aforementioned models (since γ is fixed, this tuning is with respect to λ). At this stage, we also select the optimal modeling structure. Third, we use the optimally tuned model within each modeling structure to predict outcomes on the test set $\mathbb{X}_{\text{test}}$, comparing performance between the models and confirming that the optimal structure we selected in the prior step performed the best on $\mathbb{X}_{\text{test}}$. Finally, we re-fit and re-tune the optimal modeling structure using the full data $\mathbb{X}$ in order to interpret the best final model.

In order to assess predictive efficacy for cross validation, we use the expected extra-sample Cox partial deviance, estimated as described in the `ncvreg` documentation (Breheny and Huang, 2011). While this measure is difficult to interpret in an absolute sense, it should be effective in assessing predictive accuracy in a relative sense. We calculate this predictive efficacy measure for the test data set as well.

As an additional assessment of predictive performance, we categorize individuals from the test data into three categories based on their expected risk score (low-risk, medium-risk, or high-risk). The cut points are set to be the 33rd and 67th percentile of the linear predictions on the test set, which could vary from model to model. Then, using the test set, we plot KM curves for each modeling structure that are stratified by the models' risk score categories. More separation among those stratifications on the KM plot means better predictive performance; such delineation indicates that the model is doing a good job of classifying high-, medium-, and low-risk patients.

2.4.3 Results

The estimated extra- and out-of-sample Cox partial deviance by model is shown in Figure 2.14 and Table 2.1. We find APL performed quite poorly, which seems to indicate that the consideration of pairwise interactions, without accounting for the ranked sparsity, offers negligible benefit. The LS method performed

Table 2.1: Estimated predictive performance (measured by the Cox partial deviance) on extra- and out-of-sample data broken down by modeling framework. CV refers to the average of 10-fold cross validation with 5 repeats.

	CV	Test Set
Lasso with only main effects (LS0)	10.409	7.595
SRL with only main effects, proportional weights (SR0)	10.342	7.378
SRL with interactions, proportional weights (SR1)	10.362	7.388
SRL with interactions, cumulative weights (SR2)	10.333	7.381
All-pairwise lasso (APL)	10.451	7.680

only slightly better, which indicates that penalizing the genetic and clinical covariates equally may not be advised either. The relatively strong performance of SR0 indicates that the sparsity-ranked lasso achieves a satisfactory middle ground. Since the SRL models all perform similarly, there is little to no evidence that any prominent gene-environment interactions are capable of being discovered.

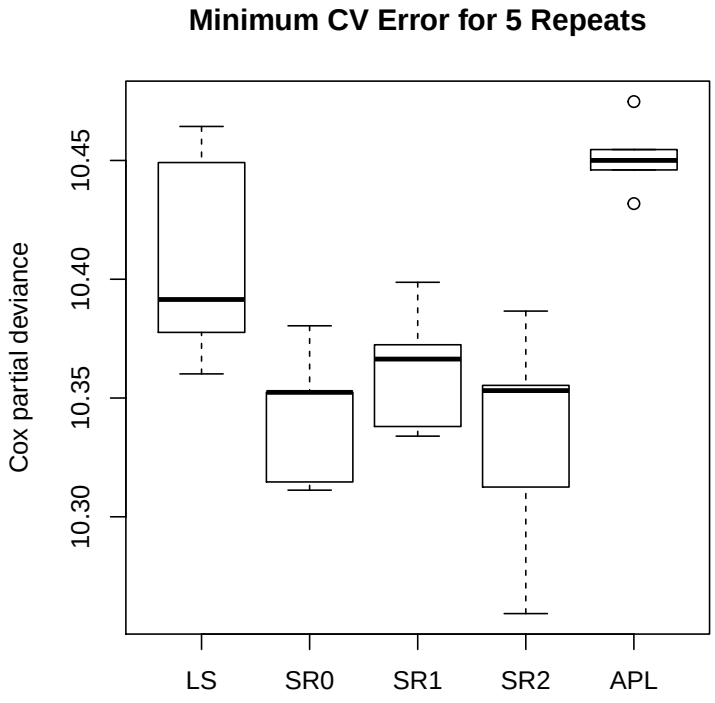

Figure 2.14: Minimum cross-validation error (estimated Cox partial deviance) for 5 repeats of 10-fold cross-validation. The spread of these results are due to the seed set prior to analysis, and do not represent the variability in the CV estimate itself.

In Figure 2.15, we show the categorization efficacy of each model using the test data set. SR2 is omitted here because its performance is very similar to SR0. In the plots, we note that the LS and the APL models did a good job classifying those with a high risk, but did not distinguish well between medium and low risk patients. The SR0 and SR1 models seem to have done a remarkable job classifying individuals in the test set, which is most likely due to the handling of the clinical covariates (SRL is shrinking the clinical variables relatively less than the LS model, and selecting fewer genetic variables).

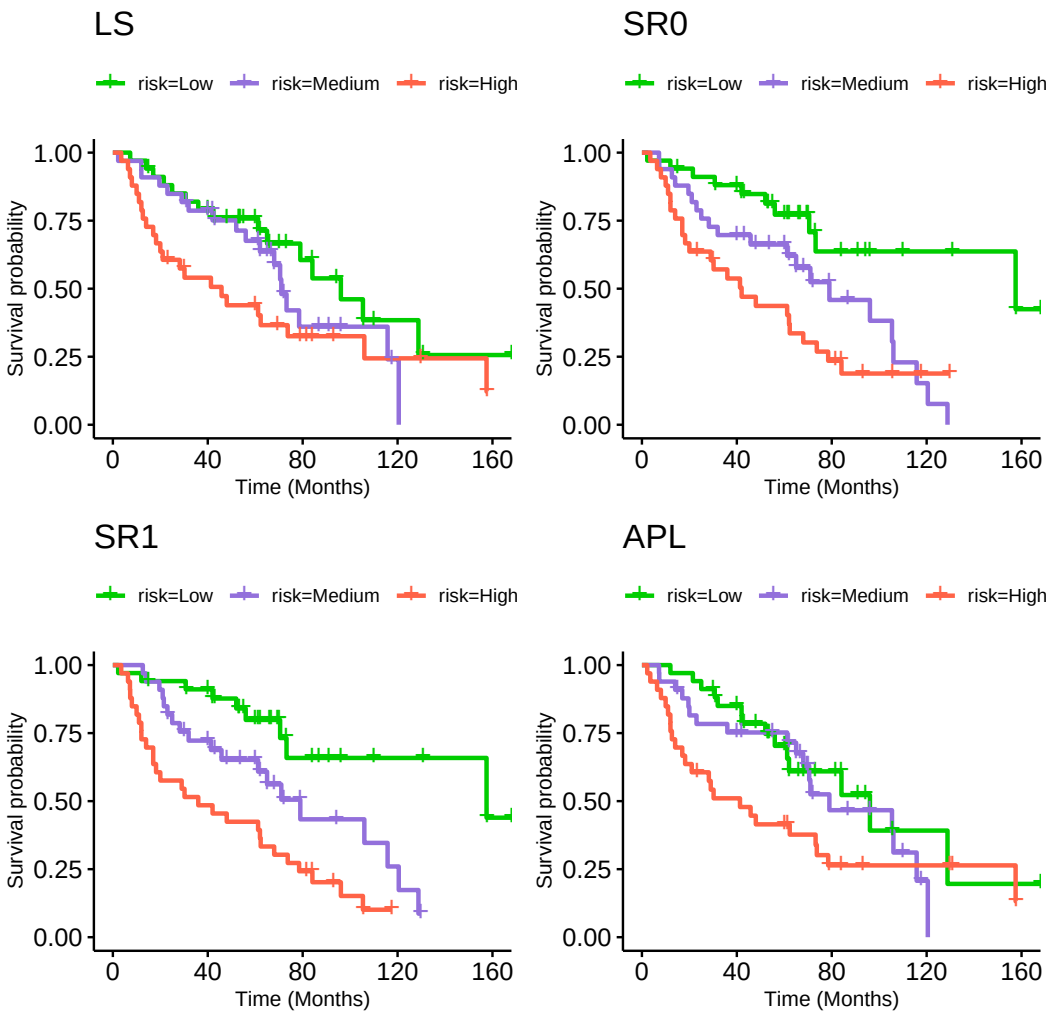

Figure 2.15: Risk score classification performance efficacy of each modeling framework using the test data set. More separation among stratifications on the KM plot signifies indicates that the model is doing a good job of classifying high-, medium-, and low-risk patients. LS refers to the lasso on the original covariates, SR0 refers to the SRL with proportional penalties on clinical and genetic covariates, SR1 refers to the SRL with interactions and proportional penalty weights, and APL refers to the all-pairwise lasso.

Table 2.2: The number of selections (S) and the sum of the magnitude of the standardized coefficients by covariate group for each (optimally tuned) model. Tuning of λ was accomplished with 10-fold cross-validation with 5 repeats, and γ was set to 0.5. LS refers to the lasso on the original covariates, SR0 refers to the SRL with proportional penalties on clinical and genetic covariates, SR1 refers to the SRL with interactions and proportional penalty weights, SR2 refers to the SRL with interactions and cumulative penalty weights, and APL refers to the all-pairwise lasso.

	LS		SR0		SR1		SR2		APL	
	S	$\|\|\beta\|\|_1$	S	$\|\|\beta\|\|_1$	S	$\|\|\beta\|\|_1$	S	$\|\|\beta\|\|_1$	S	$\|\|\beta\|\|_1$
Clinical	3	0.140	6	0.921	6	0.893	6	0.922	1	0.017
Genetic	43	1.203	39	0.830	16	0.388	39	0.854	6	0.233
Env-Env	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000
Gene-Env	0	0.000	0	0.000	0	0.000	0	0.000	4	0.060

After re-fitting and re-tuning each model to the entire data-set (again using repeated CV to calculate the optimal λ value), we can examine the number of selections (S) and the sum of the magnitude of the standardized coefficients by covariate group for each (optimally tuned) model in Table 2.2.

Evidently, the LS model found most of its signal from the genetic covariates; 43 of which had nonzero coefficients. Only 3 clinical covariates were selected, and the magnitude of the combined standardized signals ($||\beta||_1$) was only 0.14. SR0, on the other hand, found the majority of the signal to lie in the six clinical covariates ($||\beta||_1 = 0.921$), though it still found a good amount of signal (0.83) in 39 of the genetic covariates. Neither SR1 nor SR2 selected any gene-environment interactions. SR1 found less signal in the genetic variables than SR0 – this indicates that the addition of interaction terms necessitated a higher amount of shrinkage in the main effects. For SR2 however, since the coefficients are being penalized in a cumulative fashion, the amount of signal is very similar to SR0 when no interactions were considered. APL, although having discovered 4 gene-environment interactions, is clearly not able to find much signal at all; it is shrinking all of the effects considerably. All of these results taken together indicate that there are no informative gene-environment interactions, and that the clinical variables should be penalized proportionally less than the genetic variables.

In Table 2.3, we show how many selected variables were shared for each model selection method. There was very high agreement in the SR models, and in fact perfect agreement between SR0 and SR2 (they selected all of the same variables). SR1 selected only one covariate that was not selected by SR0 or SR1, a gene called

Table 2.3: Number of selected coefficients common among each method.

	LS0	SR0	SR1	SR2	APL
LS0	46	33	13	33	7
SR0		45	21	45	7
SR1			22	21	6
SR2				45	7
APL					11

"checkpoint kinase 1", although its coefficient was very small.

Finally, we can interpret the best model (SR0), which was very similar to the SR2 model. The most protective effect we found was for those in the "never smoked" group (HR = 0.73). We found two clinically significant protective gene expressions: FAM117A (HR = 0.89), and CTAGE5 (HR = 0.91). We found harmful effects if subjects were white (HR = 1.18), male (HR = 1.33), or had chemotherapy (HR = 1.93). Additionally for every 10 year increase in age, the hazard increases by a factor of 1.44. Interestingly, the clinical coefficients in this model are similar to the estimates from the model with only clinical covariates. A bit disconcerting is the fact that the identified "important" gene expressions selected by SR0 were not used as classifiers in the original paper. Note that since this was not a randomized controlled trial, these effects are not necessarily indicative of a causal relationship; the high HR on chemotherapy status is not an indication that chemotherapy is harmful.

2.5 Discussion

2.5.1 The `sparseR` Package

Since the SRL has been developed as a variant to the lasso, there are several off-the-shelf R packages that can do its "heavy lifting": `glmnet`, `ncvreg`, and `biglasso`, are viable options. Among these, we have found that `ncvreg` works well because it does not normalize penalty weights. The `sparseR` package works in concert with `ncvreg` (Breheny and Huang, 2011) to implement and facilitate SRL methods. The package also contains a suite of other ranked sparsity methods and functions, such as an information-criterion based metric that we call RBIC. A more detailed tutorial for using this package can be made available upon request;

however, the package is still in development.

2.5.2 Strengths and Weaknesses of the SRL

We have shown that the sparsity-ranked lasso performs relatively well for selecting transparent models. Whereas other methods for selecting polynomials and/or interactions tend to select overly opaque models (models with high-order relationships that are difficult to interpret), SRL naturally selects models that have more main effects and fewer "complicating" terms. In other words, the SRL limits the tendency for the lasso to select too many interactions, and controls the false discovery rate among interactions to be close to the same or less than the false discovery rate in the main effects. Therefore, the SRL is a technique that can be utilized and trusted to select from interactions and polynomials of any order.

Since many authors have already contributed to the problem of selecting from all possible interactions, discussion of the SRL method compared to the current standard is warranted. One major benefit to the SRL is that it can be applied to survival outcomes; at the time of writing, all of the other methods are supported by open-source software packages, but none can handle survival outcomes (they are all limited to binomial or continuous outcomes). Thanks to the versatility of the `ncvreg` package, the SRL method can be used for binomial, continuous, survival, or Poisson outcomes (Breheny and Huang, 2011).

Another factor to consider is the computational speed; glinternet has been shown to be 10-10000 times faster than hierNet, and yet our method is quite a bit faster than glinternet (at least for our simulation settings). This improvement in speed does not seem to change as the sample size increases, and it is especially noticeable when cross-validation is employed to tune the models. We have found that SRL works better than glinternet (in terms of prediction accuracy and the false discovery rate) when there are no interactions or when interactions are especially sparse. We have omitted some results pertaining to different hierarchical setups of the simulations in this paper, but we found that glinternet can perform slightly better when the true model is strongly hierarchical, whereas the SRL method performs better when the true model is weakly or anti-hierarchical. Their relative predictive performance depends to an extent on the mix of strong, weak, and anti-hierarchical active interactions, and more research is needed to determine exactly how and why this is the case.

Another benefit to the SRL is its straightforward extension to examine higher order interactions and polynomials; the SRL setup makes it convenient and worthwhile to investigate whether the inclusion of higher order interactions and polynomials is warranted. Further, SRL methods does not inflate the number of Type I errors in the course of this investigation.

One large weakness to the SRL is that it requires storage of a potentially large matrix of interactions into random access memory. However, with advances in the scalability of regularization algorithms such as the `biglasso` package (Zeng and Breheny, 2017), it is possible that this will become less of an issue in the future. Another weakness that the SRL shares with other regularization procedures is that the mechanism of formal inference is unclear. It should be possible to extend recent advances in the marginal false discovery rate (mFDR) (Breheny, 2018; Miller and Breheny, 2019) into the SRL framework, and this work may eventually be included in the `sparseR` package in a future update. However, for now the best mechanism for formal inference remains an area of future research.

2.5.3 Conclusion

The sparsity-ranked lasso is an effective and fast approach for selecting from derived variables such as interactions or polynomials. Ranked sparsity via the SRL implements a broader definition of Occam's Razor where a model's simplicity is not purely equated to parsimony; it is also tied to the model's transparency and intepretability. As opposed to other methods of interaction selection, the SRL does not select an unreasonable number of false interaction effects and it does not overly shrink the main effects.

CHAPTER 3
EXTENSIONS OF RANKED SPARSITY

3.1 Introduction

Chapters 1 and 2 of this dissertation motivated and explored the use of the sparsity-ranked lasso for cases when prior informational asymmetry is present among candidate predictors, focusing specifically on selecting from all pairwise interactions and polynomials. In this chapter, we touch on several extensions to the concept of ranked sparsity. First, we motivate *ranked cost models* and offer a proof-of-concept simulation study which shows how SRL can be used to optimize for prediction and data collection costs simultaneously. Second, we investigate the effectiveness of the SRL in selecting features in a time series regression framework, with an application to emergency-room hourly visit forecasting. Finally, we conclude with an extension of SRL into "learn-turn-burn" modeling which can be applied for precise and stable flu-nowcasting using a rich source of smart-thermometer data. We intend for each of these extensions to be a seed of future methodological research topics.

3.2 Ranked Cost Models

3.2.1 Background

The ranked sparsity paradigm can be useful in settings where there is a differential in terms of the *costs* of data collection. As "not all covariates are created equal" is the mantra for ranked sparsity, "not all covariates are collected with equal ease" is the mantra for ranked cost models. The phenomenon of differential data costs is an often overlooked problem in statistics and model selection. When modeling an outcome, typically it is taken for granted that future data will consist of the same potential features as the training data, and there is no traditional need to consider the fact that some covariates are easier to collect than others. However, when making predictions on new data with a relatively sparse model, such as a model fit via the lasso, why would there be a need to collect the inactive features that are not useful for prediction?

For an illustration, say we are modeling the risk of a tumor being cancerous using a collection of data sets: a feature set from a surgical procedure which removes part of the tumor for testing, imaging covariates,

genetic markers, and clinical covariates such as smoking history. These different types of data have very different collection costs. If a model selection framework suggests that the data collected surgically contain the most important risk factors for cancer status, it stands to reason that the surgery is of utmost importance to perform on new patients. After all, if we do not collect this data, how can we know that an optimally precise prediction can be made? Sure, the collection of surgical data will cost future patients money, time, pain, and it may even put them at significant risk of other complications. But the prediction will be as accurate as possible.

It suffices to say that in optimizing for the objective prediction accuracy, statisticians tend to disconnect from the subjective, human side of the equation. Unfortunately, the impact of this disconnect can be massive and detrimental to patients' health and well-being. We argue that the collection costs of future data can and should be considered during the model selection process, and one way of operationalizing this principle is with the sparsity-ranked lasso. In our motivating cancer example, we could perhaps find that there are high correlations among the surgical data and the imaging covariates. In this case, a model with imaging covariates could do fairly well at predicting cancer status while also avoiding the need to collect the surgical covariates. Even if the predictions suffer slightly, would this not a preferable model? Should statisticians always quest for the optimal predicting model, or should we be more concerned that our models will be minimally invasive to future patients?

With this motivation, we explore the use of the sparsity-ranked lasso in implementing ranked cost models in a simulation study, which serves as a proof-of-concept for this methodology.

3.2.2 Proof-of-concept Simulation

Our simulation setup takes the sample size $n = 200$, the number of features $p = 20$, and the number of active features $s = 5$. The true coefficients (indexed by j) are set to be $\boldsymbol{\beta}^T = \{1, 2, 3, 4, 5, \mathbf{0}^T\}$. Then, we run our simulations (indexed by ς) as outlined below.

For $\varsigma \in \{1, 2, ..., 10,000\}$ we complete the following steps.

- Generate $X \sim N(\mathbf{0}, \Sigma)$, where we investigate two possible Σ configurations, described subsequently. In either setting, Σ has a correlation parameter $\rho_\varsigma \sim \text{unif}(0, 1)$.

- Generate $\boldsymbol{y} = X\boldsymbol{\beta} + N(\mathbf{0}, 5^2)$.
- Generate the cost of the covariates $c_j \sim N(15, \kappa_\varsigma^2)$, where $\kappa_\varsigma \sim \text{unif}(.25, 5)$.
- Fit a lasso model without accounting for differential variable costs, and fit a sparsity-ranked lasso model with penalty weights $w_j = c_j^\gamma$.
- Determine the optimal regularization parameter λ for both models using BIC. Also, use BIC to select an optimal γ for the SRL from $\gamma \in \{0, .25, .5, .75, 1\}$.
- Generate $n^* = 10,000$ new observations from the generating distribution ($\boldsymbol{y}^* = X\boldsymbol{\beta} + N(\mathbf{0}, 5^2)$), and calculate the out-of-sample R^2 (see below) using the predictions from the lasso and SRL models ($\hat{\boldsymbol{y}}^*$).

$$R^2 = 1 - \frac{\sum_{i=1}^{n^*}(\hat{y}_i^* - y_i^*)^2}{\sum_{i=1}^{n^*}(y_i^* - \bar{y}^*)^2}$$

- For each model, calculate the cost per prediction: CPP $= \sum_{j=1}^{p} \mathbf{1}(\hat{\beta}_j \neq 0) * c_j$. Each model selects some coefficients to be nonzero, and these are features that must be collected for new data in order to make a prediction; the sum of the costs of the nonzero coefficients thus represent the cost per future prediction under each model.

For the first simulation setting, the correlation among all of the predictors is equal to ρ_s. This means that Σ is compound symmetric with correlation ρ_s. Since such a correlation structure is somewhat unrealistic, we also investigate a situation where there is only correlation among some of the predictors. Therefore in the second setting, the correlation matrix is block diagonal, such that

$$\Sigma = \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathrm{I} \end{bmatrix}$$

where Σ_1 is a 10 x 10 compound symmetric matrix with correlation ρ_s. This ensures that the signal parameters are correlated with some noise parameters, but not all of the noise parameters.

The naive results are presented in Figure 3.1.

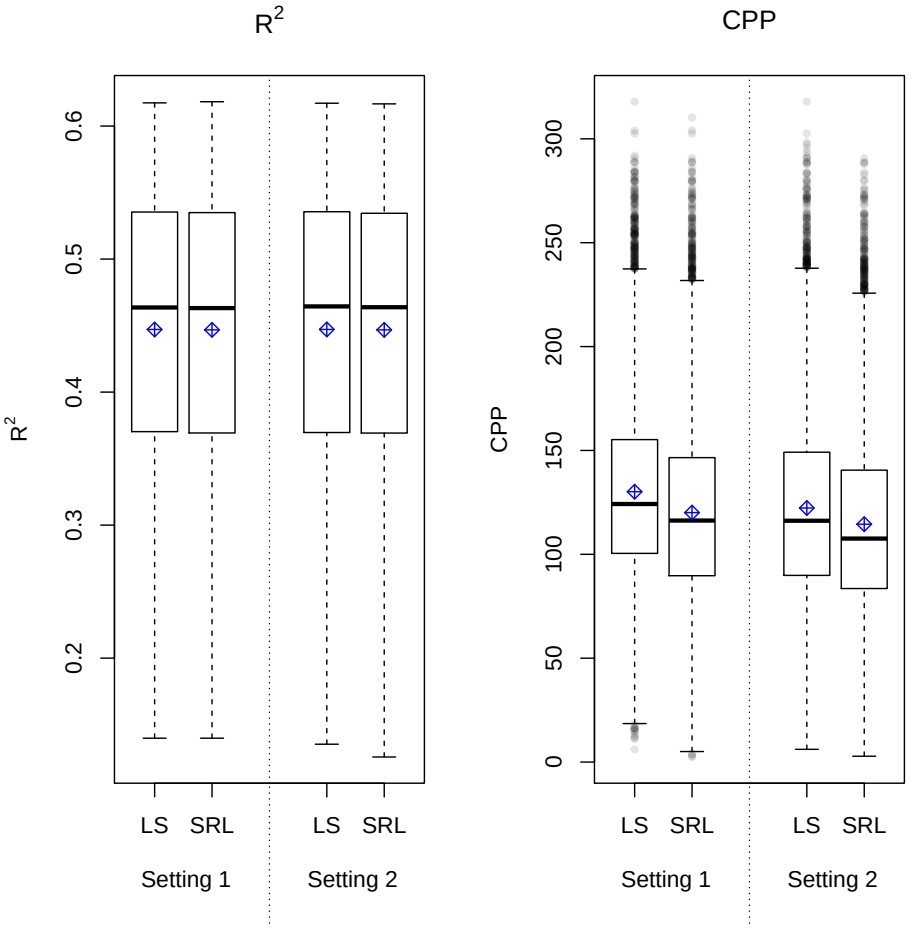

Figure 3.1: Distribution of R^2 and cost per prediction (CPP) across simulations for both simulation settings. In Setting 1, all covariates are correlated; in Setting 2 only a subset of covariates are correlated.

After these simulations were run, we fit two "meta" linear regression models to examine the SRL and LS models' relative performance with respect to R^2 and the expected cost per prediction. These meta models are structured as follows:

$$\log\left(\frac{R^2_\varsigma(\text{SRL})}{R^2_\varsigma(\text{LS})}\right) = \theta_{10} + \theta_{11}\kappa_\varsigma + \theta_{12}\rho_\varsigma + \varepsilon_{1\varsigma}$$
$$\log\left(\frac{\text{CPP}_\varsigma(\text{SRL})}{\text{CPP}_\varsigma(\text{LS})}\right) = \theta_{20} + \theta_{21}\kappa_\varsigma + \theta_{22}\log(1-\rho_\varsigma) + \varepsilon_{2\varsigma}$$

Here, we assume $\varepsilon_{1\varsigma} \overset{iid}{\sim} N(0, \sigma_1^2)$ and $\varepsilon_{2\varsigma} \overset{iid}{\sim} N(0, \sigma_2^2)$. The preceding models lead to inference based on the t-distribution for paired differences of log R^2 (or CPP) values between the SRL and LS methods, except that the models feature two additional covariates. We can interpret e^{θ_0} as the expected factor increase

(or decrease) in R^2 (or CPP) if the SRL is used instead of LS, if there is no cost variance and there is no correlation among covariates. The other parameters are slightly more difficult to interpret directly, so we show the relationships they represent using figures. The results of these meta models can be shown in Figures 3.2 and 3.3. Note that the functional forms of each model were informed by spline fits, which are also shown in these figures.

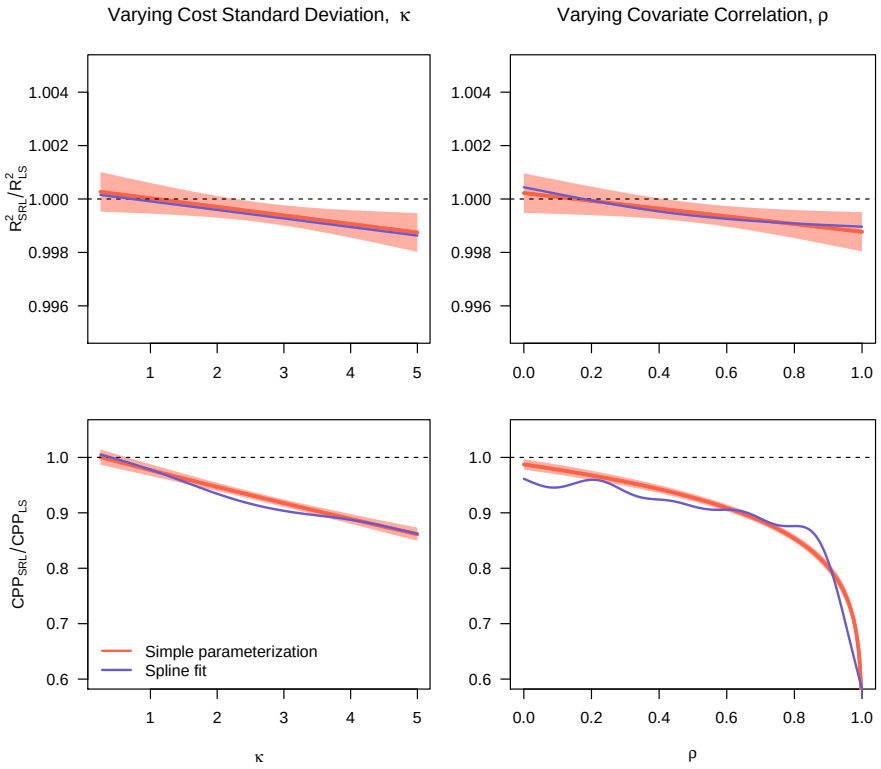

Figure 3.2: Simulation results for fully compound symmetric setting (Setting 1). Top plots represent the ratio of out-of-sample R^2 between the SRL and the LS methods; bottom plots represent the ratio of cost-per-prediction (CPP) between the SRL and LS methods. ρ is the amount of correlation among predictors, and κ is the standard deviation of covariate costs. Blue lines represent splines, red lines represent the parameterization we selected, and the shading represents a 99% CI.

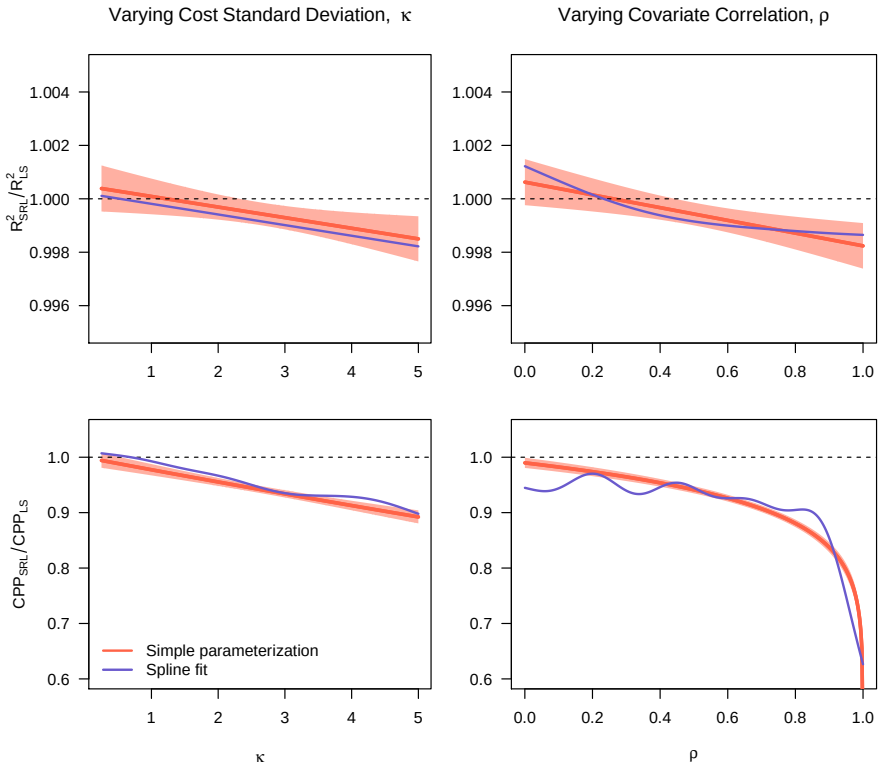

Figure 3.3: Simulation results for blocked compound symmetric setting (Setting 2). Top plots represent the ratio of out-of-sample R^2 between the SRL and the LS methods; bottom plots represent the ratio of cost-per-prediction (CPP) between the SRL and LS methods. ρ is the amount of correlation among predictors, and κ is the standard deviation of covariate costs. Blue lines represent splines, red lines represent the parameterization we selected, and the shading represents a 99% CI.

Evidently, there is a difference in the predictive efficacy for the SRL model compared to the LS model, although this difference is very small. If there is no correlation among covariates, and if there is zero variance in the covariate costs, the out-of-sample R^2 in the SRL method compared to the LS method is about 0.11% higher. It may seem odd that the SRL sees any improvement at all in this setting, but once there is any correlation or variance in the covariate cost, this improvement vanishes. For every 1 unit increase in the standard deviation of covariate costs, the expected R^2 ratio decreases by 0.03%, controlling for the correlation between predictors. We also see a negative association between the correlation among predictors and the R^2 ratio; for every 0.1 increase in ρ, the expected R^2 ratio decreases by 0.01%. In short, when the SRL begins to select slightly different variables than LS, either as a result of weighting coefficients differently by cost or high correlation among variables, its predictive accuracy suffers, albeit very slightly.

If we perform inference based on the t-distribution for paired differences in log R^2, we see that using SRL decreased the R^2 by an average of 0.05% (99% CI: [0.01, 0.09]). While these differences in the R^2 ratio are miniscule, we do see large differences in the expected cost per prediction between the two modeling methods. For CPP, the paired inference shows that the SRL decreased the CPP by an average of 10.8% (99% CI: [10, 11.6]). These paired inferential procedures are based on the average across simulations, which constitutes a case where there $\rho \approx .5$, and $\kappa \approx 2.7$. As κ increases, i.e. as there is more variability in the cost of the covariates, the SRL model is able to produce predictions at an increasingly lower cost than the LS model. Similarly, as the correlation among covariates increases, the SRL model is able to lower the cost per prediction substantially, and this benefit improves at an increasing rate as ρ increases. The results are very similar for both simulation settings; see the bottom plots of Figures 3.1 and 3.2 for a better understanding of these relationships. Depending on the κ and ρ, SRL models could produce models that predict almost as well as LS models while decreasing the cost of future predictions by 10% in the average setting to up to 40% in the highly correlated setting, and even more if there is a high amount of variance in the covariate costs.

3.2.3 Discussion

In the SRL modeling framework for ranked covariate costs, we are not looking to optimize for prediction, nor are we looking to minimize the cost necessarily. We are essentially subjecting our optimization procedure to a set of prior thoughts and beliefs about the covariates that correspond to practical concerns. In this framework, we come quite close to optimal predictions while substantially lowering the cost of making future predictions.

One major drawback to this SRL approach is the fact that it requires knowledge and/or assumptions about the cost of collection of each of these covariates. Costs are also potentially subjective, in which case they would not be truly uniform across the test data set's predictions. However, under a framework where there are defensible objective (or agreed-upon) costs to the collection of varied features, SRL provides a powerful framework for simultaneous model selection and cost minimization. It is most powerful in circumstances where there are high correlations among predictors and/or a high differential in the candidate predictors' costs.

Particularly, SRL could serve as a useful paradigm for point-of-care decision support tools. Consider, for the sake of illustration, a decision support tool for Parkinson's disease progression prediction aimed at clinicians and patients, similar to the one created by Peterson et. al for a data challenge (see https://ph-shiny.iowa.uiowa.edu/rpterson/PD_predictor/). In order to use such a tool to obtain a prediction specific to a particular person, one must input values of all of the covariates that were present in the training models. If the values are left blank, the missing features are imputed using 5-nearest-neighbors. These features which need to be input or imputed consist of a wide variety of imaging, genetic, and survey measurements. Is it fair to require patients to gather all of this data, even if it's not feasible for financial or practical reasons? Is it statistically reasonable to impute this data and take it to be a *known* value in making the prediction? We believe the answer to both of these questions is no.

With ranked cost models, a patient or clinician could enter patient-specific costs into the decision support tool that correspond to the collection of these different features. On the back-end of the decision support tool, the model could be quickly re-fit using the SRL under these specific cost constraints, thus adapting to be the best possible fitted model for that patient. If a particular piece of data cannot possibly be collected for a patient, the cost for that variable is essentially infinite and the model would adapt to not being able to incorporate that information. This essentially addresses both of the issues in the prior paragraph simultaneously.

We have shown that the SRL can be effectively utilized to produce predictive models under cost constraints, and we explored and described the trade-off between predictive accuracy and cost minimization. In a wide array of settings, the SRL can produce large benefits by reducing the cost of future data collection, while hardly changing the prediction accuracy of the fitted model.

3.3 Time Series Regression

3.3.1 Background – Autoregressive Models

Consistent data collection is beginning to pervade our lives in unprecedented and unexpected ways. Time series data, where one variable is measured many times at regular intervals, is ubiquitous. A measurement in the present usually depends on its past (say, p) values to some extent, and this dependence can be well

captured by the autoregressive model. An autoregressive model of order p, or an AR(p) model, can be written in a form analogous to a traditional regression model and estimated using standard OLS techniques. The model has the structure

$$y_t = \beta_0 + \beta_1 y_{t-1} + ... + \beta_p y_{t-p} + \varepsilon_t$$

where after controlling for the autocorrelation in the conditional mean structure, the residuals are independent and normally distributed about zero.

One common question is the best way to select p in the AR(p) model in a given application – many techniques are possible, and some of the more common ones are outlined in the next section. Another related question is how to choose the "maximum p"; that is, the highest order p that is considered in this model selection problem (we refer to this quantity as $p_{\max}$). Many times, p is rather small, and selected on the basis of visual inspection of autocorrelation function (ACF) plots. However, this technique is imperfect, as it injects human error and the potential for overfitting into the model selection process. Further, the need for visual inspection presents a barrier to automated model selection, which is highly important in some settings (see section 3.4).

A time series may be expected to follow some pattern of seasonality; for example, online searches related to the National Football League will peak during the football season every year. For this reason, it is important that this seasonality be specified in the autoregressive model. Let β_l refer to the local parameters and β_s refer to the seasonal parameters, and say (for now) that we know the periodicity of the seasonality is m time measurements. The AR model can be written as

$$y_t = \beta_0 + \sum_{j=1}^{p} \beta_{lj} y_{t-j} + \sum_{j=1}^{P} \beta_{sj} y_{t-sm} + \varepsilon_t$$

where again, the ε_t are assumed to be independent and normally distributed. As an illustrative example, say the seasonal period is yearly, and the series is monthly, then $m = 12$. If the series is weekly, then $m = 52$, and so on. This AR model is referred as a seasonal autoregressive (SAR) model, and has parameter collections of size p and P, which denote how many local and seasonal components are to be estimated, respectively. As with the standard AR model, some model selection must take place in order to select P, and in that process a decision must be made about the maximum for consideration, $P_{\max}$.

The benefits of representing the AR and SAR models in this linear model form is that it becomes clear how least-squares (or some other estimation technique) can be used to estimate the $\boldsymbol{\beta}$ parameters. However, there are also models that cannot be easily represented by this lagged model form – for instance, those with moving average (MA) components. The now-ubiquitous autoregressive integrated moving average (ARIMA) models can handle moving average terms as well as a differencing option, which may be advisable if the time series is not stationary (Cryer and Chan, 2008). In this section, we investigate how the SRL can be used in the time series regression framework; we show that the SRL can fit AR and SAR models quickly, effectively, and accurately.

3.3.2 Existing Methods for Selecting a Model's Order

For ARIMA models, it is common practice to use a nonlinear likelihood-based estimation procedure to fit models with potentially both AR and MA terms. After a set of candidate models has been fit, an information criterion, such as AIC or BIC, can be used to select an optimal model order. This process is implemented automatically in the `forecast` package in R (Hyndman and Khandakar, 2008; Hyndman et al., 2018). In this framework, the candidate models can be fit either in a step-wise fashion or in a best-subsets fashion. In either case, we refer to this method as automated-ARIMA (AA).

If only AR and SAR models are considered, then the Least Absolute Shrinkage and Selection Operator (the lasso) (Tibshirani, 1996) can be used to estimate the coefficients and select an optimal p and P simultaneously. An important benefit to this approach is that the autoregressive components can be selected and estimated simultaneously with exogenous covariates. In the typical ARIMA selection framework, the need to select from a set of exogenous covariates (denoted by X) can complicate the model selection process considerably; neither step-wise selection nor best-subsets are feasible when the dimension of X is large, when $p_{\max}$ is large, or when the sample size n is massive.

When seasonality is suspected, many possible complications often arise. In the seasonal AR model, we had to assume that m was known and fixed, which also presumes that there are no missed measurements in the series. If either m is unknown or variable, or the missingness is simply accommodated by collapsing "gaps" in the series, all of the candidate SAR models can be potentially misspecified. Unfortunately, in many

circumstances m is neither fixed nor known; for instance if the series exhibits seasonality each month, the period must adjust to the different lengths of the months in a calendar year. In the AR framework, one potential solution is to include nearby seasonal lags as additional parameters:

$$y_t = \beta_0 + \sum_{j=1}^{p} \beta_{lj} y_{t-j} + \sum_{j=1}^{P} \boldsymbol{\beta}_{sj} \boldsymbol{y}_{t \in [t-mj-\epsilon, t-mj+\epsilon]} + \varepsilon_t \tag{3.1}$$

Since it is unknown which lags near $t-m$ are important, this approach includes all the lags near $t-m$ within ϵ that may be important in modeling y_t. If we use the lasso to fit this model, it will typically estimate some coefficients to be zero, so we can reframe this selection problem, setting $p_{\max} = mP_{\max} + \epsilon \stackrel{\text{def}}{=} p^*$. When this simplifying assumption is incorporated into the framework, the seasonal AR model may be viewed as one of many candidates lasso models, and equation (3.1) simplifies to the local AR model:

$$y_t = \beta_0 + \beta_1 y_{t-1} + ... + \beta_{p^*} y_{t-p^*} \tag{3.2}$$

The preceding assumption means that a very high order AR model could be selected, depending on the expected pattern of seasonality. However, if the primary goal is prediction, there is no large downside to selecting a high order model (other than potentially adding in a lot of noise, which hopefully can be avoided with the use of the lasso and an effective model selection criterion).

It should be noted that if cross-validation (CV) is desired for any of the regularization-based approaches, the implementation of CV in the time series setting is inherently more complicated due to the dependence among observations. Left-out observations of $\boldsymbol{y}$ must pervade the entire model matrix, and imputation must be done in order to fit the model appropriately. While the implementation is not an impossible task, it is beyond the scope of this dissertation. For our purposes, we tune the lasso (and the SRL) models using either BIC or AICc.

Finally, another popular method for fitting seasonal time series with multiple modes of seasonality is called the TBATS method (which stands for "trigonometric, Box-Cox transform, ARMA errors, trend, and seasonal components") (De Livera, Hyndman and Snyder, 2011). As with automated-ARIMA, the TBATS method can be implemented efficiently in the `forecast` package, but importantly, this software does not allow for TBATS to be used in conjunction with exogenous variables. Further, the mode(s) of seasonality

must be pre-specified in this framework.

3.3.3 Dynamic Penalty Tuning with the Sparsity-Ranked Lasso

The sparsity-ranked lasso (SRL) is designed to be of use when the assumption of covariate equipoise, defined as the prior belief that all covariates are equally worthy of entering into the model, is not satisfied. In the AR framework, this assumption is definitely not satisfied, and the SRL can step up to address the resulting challenge. Particularly, going from equation (3.1) to equation (3.2) in the previous section is suspect; we have very good reason to believe that the effects on the lags between p and $m - \epsilon$ are equal to zero. We are also much more inclined to think that more recent lags are more likely to be important *a priori* than older ones. If m is known, we are much more willing to believe the mth lag to be predictive than other lags. The SRL can accommodate such expected differences in skepticism by scaling the penalty differently for different lags in the model. In this subsection, we discuss two methods for adapting the SRL to handle time series data: we can either parameterize skepticism, or we can use the data to inform it.

Parameterizing skepticism

Since we have a good idea that recent and seasonal lags are more likely to be important, but we are not certain, we can operationalize this notion using what we call *penalty-scaling functions*. These penalty-scaling functions, which we denote as f, provide ways of informing the weight of the lasso penalty w_j that corresponds to the jth lag, i.e. $w_j = f(j, \cdot)$. We discuss two different penalty-scaling functions: one that relates to a lag's "age" (i.e. how recent it is), and one that relates to its season. We also show how combinations of these two penalty-scaling functions can be utilized that are widely applicable. For age, we use γ_a to illustrate the strength of this scaling factor, which is treated as a tuning parameter. For $\gamma_a \geq 0$, and for a coefficient on the jth lag of y_t,

$$f_a(j, \gamma_a) = (j/p^* + c)^{\gamma_a} = \exp\{\gamma_a \log(j/p^* + c)\}$$

If $\gamma = 0$, the resulting procedure is equivalent to the ordinary lasso, since the function evaluates to 1 $\forall$ j. The addition and the choice of the constant c is somewhat arbitrary. If $c = .5$, this ensures that the scaling factor evaluates to 1 at the middle lag considered, $p^*/2$. This choice of c effectively relaxes the original lasso's

penalty for early lags, and increases it for later lags. One could also set $c = 1$ instead, which would simply increase the penalty for all lags instead of having the relaxing property. In practice, since one has to tune the original lasso penalty λ anyway, the choice of c typically makes little to no difference. This function is visualized with $c = .5$ for various γ_a in the top plot of Figure 3.4.

The seasonal penalty-scaling function is slightly more complicated. The parameter we use to illustrate the strength of this function is γ_s. For $\gamma_s \geq 0$, the penalty scaling function for the jth lag is

$$f_s(j, \gamma_s, m) = \exp\left\{\gamma_s \left[-\cos\left(\frac{2\pi * j}{m}\right)\right]\right\}$$

Here m is the (suspected) seasonal frequency, which is set to 52 in the visualization in the middle plot of Figure 3.4. Similar to the age function, setting γ_s to 0 yields the ordinary lasso. Finally, we can combine the two penalty-scaling functions by simply multiplying f_a and f_s together. For $\gamma_a \geq 0, \gamma_s \geq 0$, we have

$$\begin{aligned} f_{as}(j, \gamma_s, \gamma_a, m) &= \exp\left\{\gamma_s \left[-\cos\left(\frac{2\pi * j}{m}\right)\right]\right\} * (j/p^* + c)^{\gamma_a} \\ &= \exp\left\{\gamma_s \left[-\cos\left(\frac{2\pi * j}{m}\right)\right] + \gamma_a \log(j/p^* + c)\right\} \end{aligned}$$

Again, this penalization is easiest to understand when visualized for different values of γ_a, γ_s, and j; we show several different configurations in the bottom plot of Figure 3.4.

This SRL method is not without drawbacks. For instance, if m is completely unknown ahead of time, one cannot parameterize the seasonal scaling function correctly, and it also becomes unclear how to best select p^*. Another potential downside to this method is that the algorithm may become unstable if the "true" generating mechanism is an SAR1 model, i.e. the m is known exactly. SRL penalizes the lag-m coefficient very similarly to those surrounding m, and since the lasso is prone to select from correlated variables somewhat randomly, chances are high that the SRL will select lags nearby to m instead of m, thus reducing its accuracy relative to a traditional SAR(1) fit. Finally, the combined penalty scaling function is not always defensible; the period of the wave depends (to a small degree) on the age hyperparameter, and therefore the two γ parameters somewhat interfere with each other. This can lead to issues of identifiability; we have found in practice that the optimal solution using this technique can lie on a plane of γ_a, γ_s tuning parameters (though this plane often does not intersect with the lasso solution).

We have explored these parameterized penalty-scaling functions through simulations and we have

applied them successfully in multiple applications. However, we omit these simulations and results for the sake of brevity, and because the SRL method that we discuss in the next subsection performs similarly and is more broadly applicable than the parameterized version.

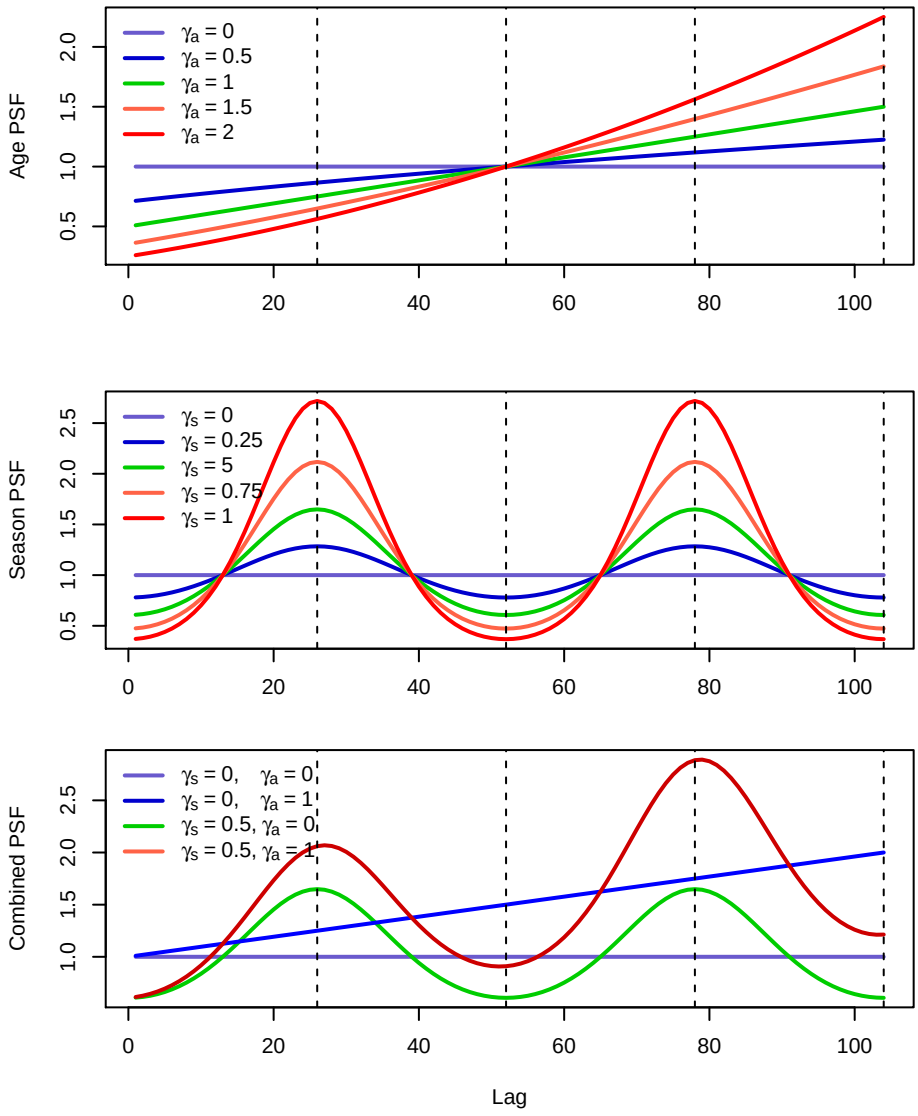

Figure 3.4: Parameterized penalty scaling functions (PSF) for coefficients on the lags in a time series SRL setting. The top plot refers to the age PSF ($c = .5$), the middle plot refers to the seasonal PSF, and the bottom plot is their combined PSF. In the bottom plot, $c = 1$.

SRL with the Partial Autocorrelation Function

Instead of having a predefined parameterized ranking of skepticism for various lags in a time series

model, we could use the data to inform these penalty weights. A useful means to accomplish this is to use the Partial Autocorrelation Function (PACF), defined as the correlation between y_t and y_{t-k} after removing the effect of the intervening variables $y_{t-1}, y_{t-2}, ..., y_{t-k+1}$. The PACF is easy to calculate, and it provides a data-driven measure of the importance of each lag – many practitioners inspect the PACFs to determine which lags should be considered and estimated in a model (Cryer and Chan, 2008).

Recall the adaptive lasso (Zou, 2006), where penalty weights w_j are informed by a consistent initial "first-stage" estimate of $\boldsymbol{\beta}$:

$$||\boldsymbol{y} - X\boldsymbol{\beta}||^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|$$

Since the PACFs are in some sense first-stage estimates of our coefficients on the lags, this SRL and PACF procedure is very similar to the adaptive lasso, where

$$w_j = \left(\frac{1}{|\hat{\phi}_j|} \right)^{\gamma}$$

Here $\hat{\phi}_j$ represents the estimated PACF on the jth lag of y_t. Importantly, ϕ_j can be written as the parameter in an AR(j) model. Since we can formulate this AR(j) model and estimate ϕ_j for any j with least squares, we know that $\hat{\phi}_j$ is a consistent estimator. Therefore, this procedure can enjoy the same oracle property as the adaptive lasso. If there are exogenous covariates, then some additional weights can be applied to these as well; we suggest setting the weights on exogenous covariates to the marginal OLS estimates or the OLS estimates controlling for some of the autocorrelation in the series. We refer to this approach as the SRLPAC (sparsity-ranked lasso with partial autocorrelation) procedure, pronounced "SRL-pack".

The SRLPAC approach is ideal for modeling time series data with complicated seasonality; it is quick, intuitive, and it it can be conveniently tuned using AICc or BIC provided the sample size is large, which is typically the case in time series settings. Further, SRLPAC does not require the pre-specification of seasonal modes at all; it does this naturally and automatically using the PACF function. Finally, the SRLPAC approach allows for seamless integration of exogenous variables.

3.3.4 Measuring Predictive Accuracy

In order to measure prediction accuracy in a time series setting, there are many possible options. In this chapter of the dissertation, we utilize several different methods, each of which is focused on the predictive accuracy of the models in forecasting new data. We use the root-mean-squared prediction error (RMSPE), R^2, the mean absolute error (MAE), and the mean absolute percentage error (MAPE) (Hyndman, 2006). These metrics are all estimated using out-of-sample data, and are subsequently defined in this section.

For a set of predicted values $\hat{\boldsymbol{y}}^*$, and a set of n^* out-of-sample observations that were not used in calculating these predictions, $\boldsymbol{y}^*$, we define RMSPE as

$$\text{RMSPE} = \sqrt{\frac{1}{n^*}\sum_{i=1}^{n^*}(\hat{y}_i^* - y_i^*)^2}$$

RMSPE is intepretable in the same unit of measurement as the original measurement for y, and it signifies how far away from the true value our predictions were, on average. Another measure that we already used in the first section of this chapter, R^2, is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n^*}(\hat{y}_i^* - y_i^*)^2}{\sum_{i=1}^{n^*}(y_i^* - \bar{y}^*)^2}$$

This definition of R^2 differs slightly from other definitions in that there is no guarantee that it be nonnegative, as is the case for R^2 in its typical in-sample setting. As a result, this version of R^2 does not have quite as clean of an interpretation. However, values close to 1 indicate very accurate predictions relative to naively guessing the mean, and values close to 0 indicate that using the mean of the responses for forecasting would have done as well as the model-based predicted values. Negative values indicate that the model-based predictions did substantially worse than the mean.

Since the RMSPE metric is not robust to outliers, another commonly used metric is the mean absolute error, defined here as

$$\text{MAE} = \frac{1}{n^*}\sum_{i=1}^{n^*}|\hat{y}_i^* - y_i^*|$$

The interpretation of this measure is very similar to RMSPE. A related metric is the mean absolute percentage

error, which is defined as

$$\text{MAPE} = \frac{100}{n^*} \sum_{i=1}^{n^*} \frac{|\hat{y}_i^* - y_i^*|}{y_i^*}$$

The scaling of the summed errors by the value of y_i^* and its multiplication by 100 entail that the MAPE has a percentage interpretation: how far off, in terms of a percentage, can we expect our predictions to be from the true value, on average? In this formulation, the denominator is often replaced with $\bar{y}_i^*$ to account for zero values (a convention we follow in the subsequent sections).

3.3.5 Application – Emergency Room Visits

SRLPAC is able to shine especially when there are multiple modes of seasonality. In this application, we showcase some of these benefits in a novel approach to emergency room visit forecasting. For planning purposes, it is highly desirable to be able to accurately forecast the expected number of visits in the emergency room (ER) each hour. Accurate forecasting (and planning) reduces the costs and frustrations associated with having to "call-in" extra help. Previous research has indicated that time series models can be developed and utilized to predict ER visits on the daily time-horizon (Jones et al., 2008). However, these daily forecasts, while informative, are less helpful from a practical standpoint since shifts are typically set for under 12 hours. It is useful to have more granular predictions so that personnel resources can be allocated more efficiently on a shift-by-shift basis. Other research investigating the forecasting of ER arrivals at the hourly level does exist, and mostly utilize seasonal ARIMA models to fit the hourly series for a single ER. A detailed literature review is outside the scope of this dissertation.

In this study, we examine hourly visit counts for the Emergency Department at the University of Iowa Hospital and Clinics (UIHC) from July 2013 to March 2018. Since this is hourly data over a long time span, the sample size in this modeling problem is large ($n = 41,640$). Further, multiple modes of seasonality are feasible; we expect to see more visits during the day than at night, and there could be weekly and monthly periodicities as well. The high sample size also yields minor computational concerns for some of the methods we have described.

The PACF is shown in Figure 3.5; we see that there are indeed many modes of seasonality. The most

prominent correlation is the AR(1) term, but there are also large spikes around 23 hours, 47 hours, 72 hours, 7 days, 14 days, 21 days, and 28 days. Curiously there does not seem to be a prominent month effect around the 30-day mark.

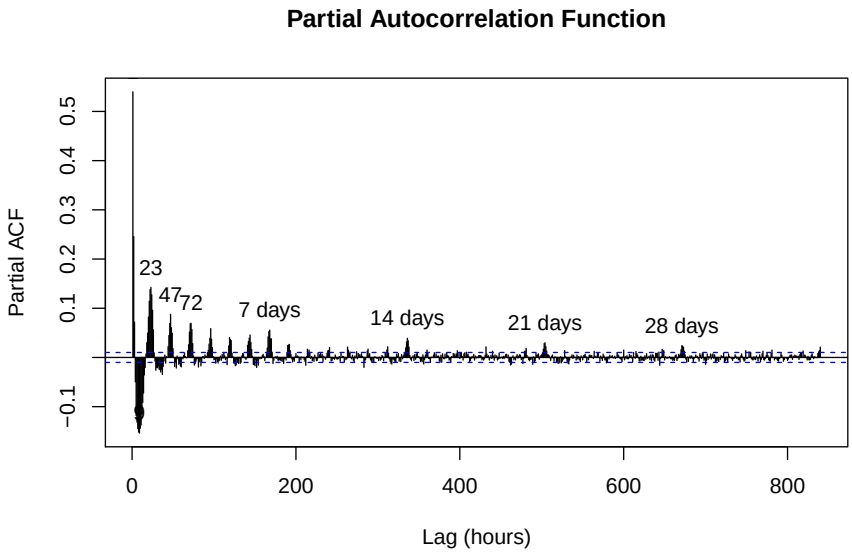

Figure 3.5: PACF function for 840 lags of hourly visits to the emergency room.

While many exogenous variables could feasibly factor into the number of emergency room visits in an hour, for the sake of this problem, we limit our consideration to monthly fixed effects, holiday effects, and concurrent hourly temperature in Iowa City, IA, as exogenous features (which are unpenalized in the regression). For holidays, we included separate fixed effects on Christmas, New Year's Eve, New Year's Day, Thanksgiving, Independence Day, and Hawkeye Gameday. We also include holiday "plus-one" effects for Christmas and Thanksgiving to examine if there are lingering effects of these major holidays.

Before fitting the models, we divide our sample into a training set (the first 90% of data, 7/1/2013 until noon 10/9/2017) and a test set (the final 10% of data, starting noon 10/09/2017 to 3/31/2018). Using the training set, we fit the following models:

1) SRLPAC tuned with AICc
2) SRLPAC tuned with BIC
3) SRLPAC tuned with AICc, with exogenous variables (SRLPACx)

Table 3.1: Model performance for hourly arrivals. Prediction metrics were estimated using a left-out test sample, and the mean computation time (MCT) in minutes was computed across 12 replications on a single core of a machine running Windows 10.

	R^2	RMSPE	MAE	MAPE	MCT (Minutes)
SRLPAC – AICc	0.528	2.651	2.051	31.3	1.00
SRLPAC – BIC	0.526	2.656	2.057	31.4	1.00
SRLPACx – AICc	0.530	2.645	2.045	31.2	1.32
SRLPACx – BIC	0.529	2.648	2.049	31.3	1.32
Auto-SARIMA	0.293	3.242	2.528	38.6	12.16
TBATS	0.520	2.672	2.075	31.7	9.66

4) SRLPAC tuned with BIC, with exogenous variables (SRLPACx)

5) Automated seasonal ARIMA (auto-SARIMA) selected using AICc. Note that for the auto-SARIMA approach, only one mode can be supplied, so we specified a daily frequency ($m = 24$) for this setting based on the PACF plot.

6) TBATS model with mid-day, daily, weekly, and monthly frequencies specified: $m = (12, 24, 168, 672)$.

After fitting these models, we then investigate the metrics outlined in the previous section. We repeat the fitting process 10 times to calculate the mean computation time (MCT) for each method in minutes. For the SRLPAC model with exogenous terms, we interpret these terms in their context. Finally, for staffing purposes, it is arguably more informative to investigate how well the model performs at predicting the number of visits that occur in a given 8-12 hour window, rather than at the hourly level. So, we also investigate the performance of the SRL and TBATS methods in predicting the 10-hour rolling sum of patients, which was calculated using the sum of the 1- through 10-step-ahead predictions on the test data set.

Results

We see in Table 3.1 the auto-SARIMA approach performed the worst by all measures; it took a relatively long time to run, and the models it produced did not predict accurately. The TBATS method did predict quite well, though it also took much longer to run than SRLPAC. SRLPAC performed admirably, both with and without the exogenous variables, the inclusion of which seems to offer a modest improvement in predictive accuracy. The best predicting model was the SRL with exogenous variables, tuned with AICc. In this optimal model, the estimated coefficients for the exogenous variables are shown in Figure 3.6 as well as summarized in the Table 3.2.

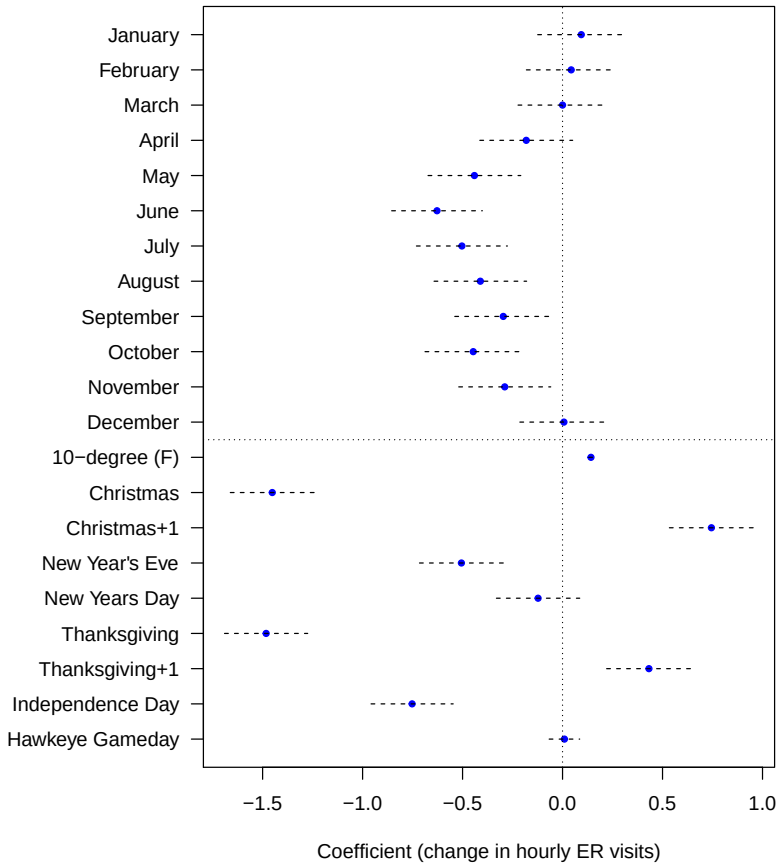

Figure 3.6: Coefficient estimates and 95% confidence intervals for exogenous variables in SRLPAC model. Values refer to the expected change in hourly visits controlling for other factors in the model, including the temporal correlation among observations.

Hourly ER visits do seem to exhibit seasonality on a yearly-scale, controlling for the rest of the covariates in the model. Specifically, there were fewer patients per hour in the summer than in the winter. It is important to realize, however, that we are controlling for the hourly concurrent mean temperature in the model as well, which has a very strong positive association with patient arrivals. For every 10 degree F increase in temperature, we expect to see about 0.15 more patients per hour, controlling for the month and holiday effects. We observe very strong negative effects on the major holidays, most of all Christmas and Thanksgiving on which there were about 1.5 fewer patients per hour, on average. This effect seems to have consequences; on the day after Christmas and Thanksgiving, there are about 0.75 and 0.4 more patients per hour, respectively. The effects of New Year's Eve, New Year's Day, and Independence Day are all negative,

Table 3.2: Estimated coefficients and confidence intervals for the SRLPACx – AICc model.

	Coefficient	95% CI
Monthly Fixed Effects		
January	0.094	(-0.12, 0.31)
February	0.043	(-0.18, 0.27)
March	Baseline	
April	-0.182	(-0.41, 0.05)
May	-0.441	(-0.67, -0.21)
June	-0.628	(-0.85, -0.4)
July	-0.503	(-0.73, -0.28)
August	-0.411	(-0.64, -0.18)
September	-0.296	(-0.54, -0.05)
October	-0.447	(-0.69, -0.21)
November	-0.289	(-0.52, -0.06)
December	0.007	(-0.21, 0.23)
Other Effects		
10°F Increase	0.142	(0.13, 0.15)
Christmas	-1.452	(-1.66, -1.24)
Christmas+1	0.744	(0.53, 0.95)
New Year's Eve	-0.506	(-0.72, -0.3)
New Years Day	-0.122	(-0.33, 0.09)
Thanksgiving	-1.482	(-1.69, -1.27)
Thanksgiving+1	0.432	(0.22, 0.64)
Independence Day	-0.752	(-0.96, -0.55)
Hawkeye Gameday	0.009	(-0.07, 0.09)

but are less pronounced than Christmas and Thanksgiving. There does not seem to be a large change in the number of patients per hour on Hawkeye Gamedays; the 95% confidence interval is quite concentrated around 0. Note that each interpretation of these fixed effects is also controlling for the "important" local and periodic autocorrelation in the series by the nature of the SRLPAC approach.

Table 3.3 shows how well each model predicted how many patients arrived in the ER in a 10-hour window. The SRLPAC methods performed similarly to one another, though the SRLPAC model selected by AICc with exogenous covariates did the best. This optimal model was able to describe 84.2% of the variation in the outcome. Its MAE was the lowest; its prediction was on average only about 7.05 patients off of the true value. Finally, Figure 3.7 shows the accuracy of the SRLPAC's model in predicting the 10-hour rolling counts.

Table 3.3: Model prediction performance for 10-hour rolling sum of patient arrivals.

	R^2	RMSPE	MAE	MAPE
SRLPAC – AICc	0.839	9.006	7.089	10.8
SRLPAC – BIC	0.834	9.145	7.214	11.0
SRLPACx – AICc	0.842	8.904	7.054	10.8
SRLPACx – BIC	0.840	8.986	7.121	10.9
TBATS	0.823	9.434	7.411	11.3

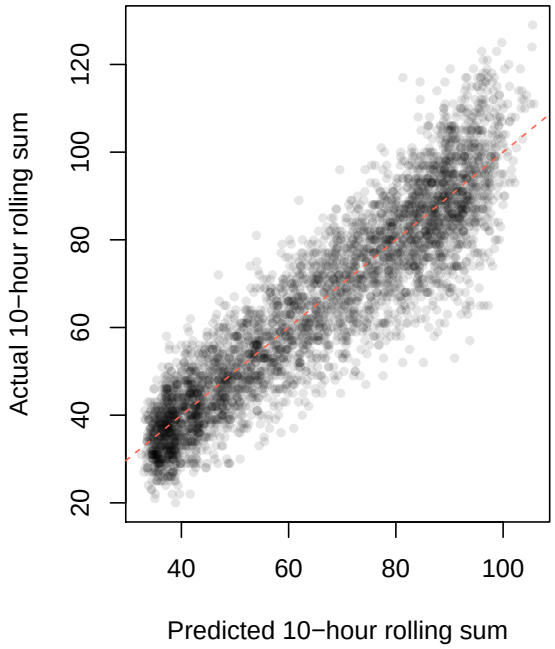

Figure 3.7: Absolute vs. predicted values for 10-hour patient arrival counts for SRLPAC model with exogenous variables tuned with AICc. The dotted line is where $y = x$.

3.3.6 Discussion

We have shown that the SRL is adaptable to time series data in several ways. Not only does the SRLPAC method rival the prediction accuracy of other state-of-the-art forecasting methods, it is able to incorporate exogenous data seamlessly and run considerably faster for large data sets. Using the model fit with SRLPAC, we were able to accurately forecast the number of patients who would arrive at the UIHC

Emergency Department to within an average of about 7 patients per 10-hour time window. We were also able to use the SRLPAC model to make inferences about patterns in the data, finding large correlations between ER arrivals and temperature, month, and holidays.

Many researchers have shown that SARIMA models are effective in forecasting the number of ER arrivals at other institutions. However, we have shown that at least for the UIHC Emergency Department, the SARIMA model does not perform effectively compared to other methods. In future work, we will compare the SRLPAC approach to other popular forecasting methods in this domain, which include SARIMAX (SARIMA with exogenous variables) and neural networks. We will also investigate model averaging approaches which blend multiple values of the tuning parameters of SRLPAC (γ and λ) based on their likelihoods to provide a potentially more accurate prediction.

3.4 Learn-Turn Models – Nowcasting the Flu using Smartphone Data

3.4.1 Background

With sparsity-based, automatic, and fast model selection tools in the time series setting, it becomes much easier to create "smart" forecasting models – models that learn new aspects of the underlying phenomenon over time, and that can dynamically adapt. A popular method in some fields (e.g. the flu forecasting literature) is called dynamic adaptive out-of-sample model fitting. We call this approach the "learn-and-burn" method. For a window of m days, a model is fit to times 1, 2, ..., m, and used to predict y_{m+1}. Once a prediction $\hat{y}_{m+1}$ is computed, the window moves one time period into the future; a model is fit to time periods 2, 3, ..., $m+1$, and used to obtain $\hat{y}_{m+2}$. This window continues to move until predictions have been obtained on all of the remaining data points. Early time points are essentially not used in the fitting of many of the final predictions, so therefore these data are in some sense "burned". Due to what ended up being a lackluster performance of the Google Flu Trends (GFT) project in predicting the flu, these adaptive approaches have become more popular. GFT attempted to use search data to predict the flu on a state and regional basis, but since it was only trained once to past data, it was unable to adapt to changes in user behavior and searches, and it was thus overfit to the past data. The learn-and-burn approach protects against this type of overfitting. Rather than completely burning past data, we have adapted the SRL to "turn" our model at each move of

the window based on where it was at prior time points. We believe that this approach will result in more stable parameter estimates over time, while still retaining the adaptivity of the learn-and-burn approach. We call this new method the "learn-turn-burn" (LTB) method.

To showcase the LTB method, we return to the flu series that challenged the predictive efficacy of GFT. Thankfully, we have an additional source of data that can improve our predictions. The flu outcome is an estimated percentage of outpatient visits that present with influenza-like illness (ILI), as reported by the Centers for Disease Control (CDC). More specific details about this measure can be found at https://www.cdc.gov/flu/weekly/overview.htm. The national ILI series over the course of our study period is presented in Figure 3.8.

The Kinsa Smart-Thermometer is a device that can be plugged into a smartphone and used to take one's temperature. Kinsa has distributed these devices across the country, and has given us access to their data. Specifically, we have state, regional, and national weekly counts of the number of readings and the number of fevers. We can tell whether each measurement was a repeat or not, so we also have the number of *distinct* readings and fevers as additional features for the state-, regional-, and national-levels. These data are available from August 26th, 2015, to April 4th, 2018.

We also obtained Google search data via Google Correlate (Mohebbi et al., 2011), through which we identified one hundred search queries that are most correlated with the ILI nationally (up until June 30, 2014). The volume of these search terms was then collected at a national level for our study period. For the purposes of this section, we restrict our study time to June 30, 2014, through April 4, 2018.

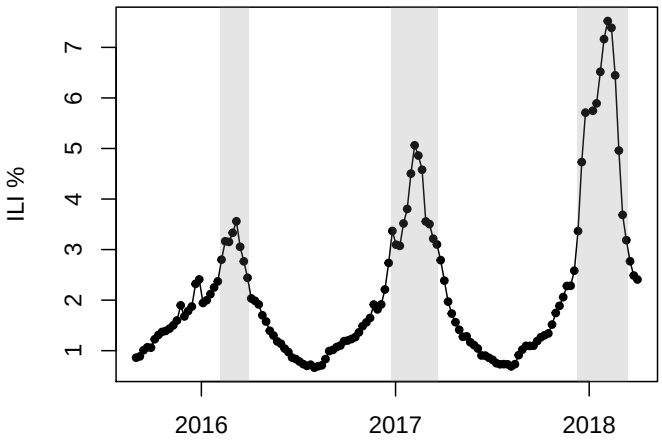

Figure 3.8: Influenza-like illness (ILI) percentage over our study period. Grey shading represents the flu season, and corresponds with the grey shading of subsequent figures.

Instead of flu "forecasting", with real-time data such as Google or Kinsa data, it is possible to "nowcast" the flu. While only a slight distinction, the application is quite different because it allows for more contemporaneous data to be used in the modeling of the outcome. Specifically, within a given learning window, the model has the form

$$y_t = \beta_0 + \sum_{j=1}^{P} \beta_{\ell j} y_{t-j} + \sum_{j=1}^{K} \beta_{\kappa j} X_{tj} + \sum_{j=1}^{G} \beta_{gj} X_{tj} + e_t$$

where the first summation represents the autoregressive structure, the second summation represents the contemporaneous Kinsa features (of which there are K), and the third summation represents the contemporaneous Google features (of which there are G). The remainder of the errors are assumed to be independent and normally distributed, while noting that the autoregressive tendency of the time series is captured by the conditional mean structure, i.e., the $\beta_{\ell j}$ parameters. In our analysis, there are 5 Kinsa features ($K = 5$), and we aggregate the Google search query volume for the 100 terms by taking their weekly mean, so the number of Google features is effectively $G = 1$. Within each learning window the covariates are standardized, so the $\boldsymbol{\beta}$ parameters are all on a similar standardized scale.

3.4.2 Turning Models with the SRL

Learn-turn-burn modeling with the SRL is highly reminiscent of SRLPAC and the adaptive lasso. First, weights are estimated for each coefficient using some initial estimator of those coefficients. In the context of the LTB method, we can use the SRL in a similar fashion where the initial estimator is not the OLS or PACF estimator, but the SRL estimate from the previous learning period. In our case, we have an additional index, m, which refers to the multiple coefficient estimates from each learning window. Specifically, we set

$$w_{jm} = \left(\frac{1}{|\hat{\beta}_{j(m-1)}| + 1} \right)^{\gamma}$$

where $\hat{\beta}_{jm}$ refers to the estimate for the previous (m) learning window provided by a lasso fit to the model specified in the previous section, and w_{jm} refers to the weight on β_j for the mth window. We have added 1 to the denominator here because we want to preclude the coefficients for a particular window from having an infinite penalty based on the time period before; we actually expect these relationships to turn on and off throughout the study to an extent, as the Kinsa variables are more informative later in the study period than early on. The Kinsa variables are in fact completely missing for the first year of this study period. It is unclear whether this formulation of the weights will lead to the oracle property, but it is also unclear whether consistency and the oracle property are meaningful concepts in this context where we do not expect n to ever be very large and where we do expect the true model to shift over time.

3.4.3 Methods

We produce three models. First, we fit the learn-burn (LB) model using the lasso with only endogenous terms, where the weights are not updated at each step; this is referred to as the LB0 model. Second, we fit a LB model with all of the exogenous and endogenous features, which is referred to the LBX model. Finally, we fit a weight-updating SRL model with all of the features: the LTB model.

In order to examine these models' adaptivity and stability, we produce trace plots for each coefficient over the course of the study. We also investigate the accuracy of the nowcasts in each setting. We tune the models using AICc, and for the first learning period the penalty weights are set to be equal for each of the variables. Prediction (nowcast) accuracy is summarized using the metrics described in section 3.3.4.

Table 3.4: Predictive performance of nowcasts for each method. LB0 refers to the endogenous learn-burn model, LBX refers to the learn-burn model with exogenous covariates, and LTB refers to the learn-turn-burn model with exogenous covariates.

	R^2	RMSPE	MAE	MAPE
LB0	0.867	0.552	0.342	15.898
LBX	0.939	0.375	0.241	11.191
LTB	0.940	0.370	0.228	10.601

Additionally, we investigate the performance of each model over the course of our study by plotting the errors by week, then fitting a kernel-based smoother on these errors for each method. This yields insight into the performance differences among models across time, and in particular whether the predictions are effective during versus outside of the flu season.

3.4.4 Results

The predictive performance results are shown in Table 3.4. Though all of the methods did reasonably well with high R^2 values, we definitely observe a strong improvement when we add exogenous variables. We see that the LBX and LTB models have quite similar predictive accuracy, though LTB slightly outperformed LBX. The LTB method seemed to improve the MAPE the most out of all of the metrics.

We can get a better sense of the predictive accuracy across time by looking at the nowcast errors as a function of time, and estimating a smooth relationship between them; these smoothed error curves are presented in Figure 3.9. We see that the LB0 model is doing relatively poorly at nowcasting during the flu season, and that both LBX and LTB models substantially improve the predictions during the flu season. For much of our study, the LBX and LTB model track very closely with one another. However, during the 2017 flu season and off season, the LTB model seems to have done slightly better in terms of the estimated absolute percentage error. Without more data, it is difficult to know whether these differences are substantial enough to show conclusive evidence that the LTB method is preferred over the LB method (if one is solely interested in predictive accuracy).

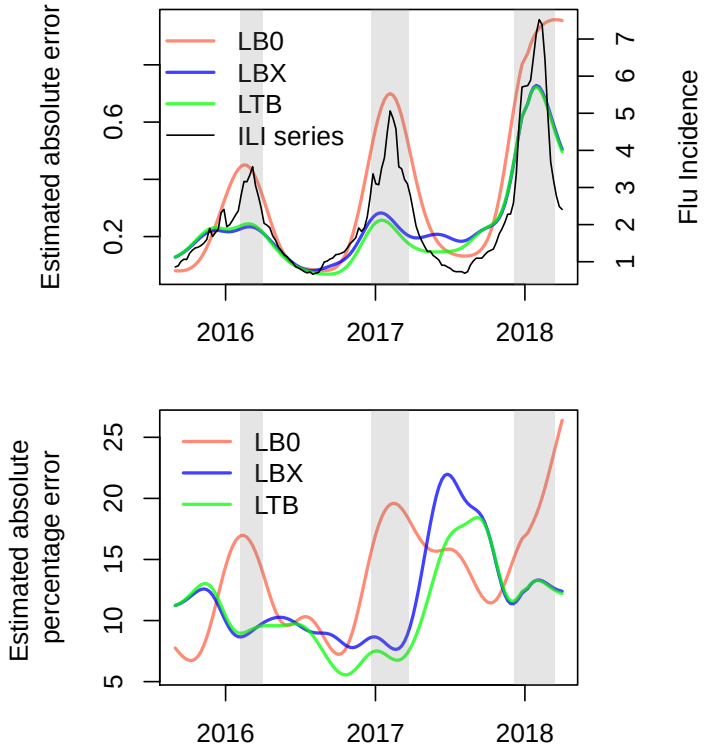

Figure 3.9: Estimated absolute error (top) and estimated absolute percentage error (bottom). Colored lines represent kernel-smoothed estimate of the expected error measurement. The top plot overlays the ILI flu series, which is on a different scale denoted by the right axis. LB0 refers to the endogenous learn-burn model, LBX refers to the full learn-burn model, and LTB refers to the full learn-turn-burn model.

The adaptiveness of the learn-burn approach leads to higher variability and less stability of the model's parameters over time. With the LTB SRL method, we can improve the model stability, so that if model stability is of value, it does not have to be completely abandoned in the pursuit of adaptivity. The trace plots presented in Figure 3.10 illustrate this stability. The green lines in these plots represent the LTB coefficients over time, and while they are still quite variable, the model is substantially less variable than the LBX method's coefficients. This stability difference is especially evident in the coefficient for "Total Readings", for which the LBX coefficient is interestingly very unstable at around the same time that the LBX model exhibited slightly worse predictive performance. This behavior could serve as evidence that the improved stability of the LTB model did indeed enhance the predictive performance in 2017.

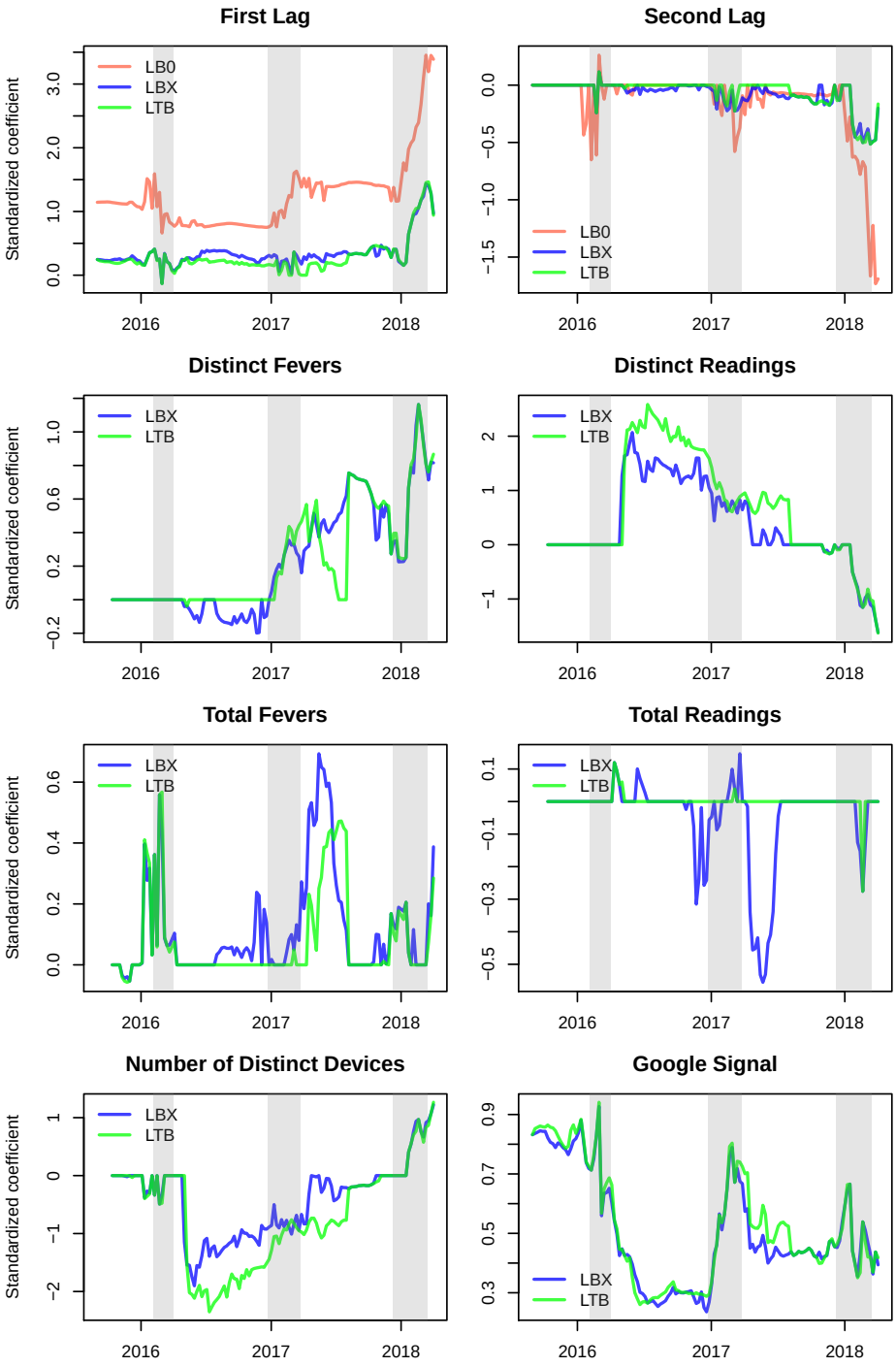

Figure 3.10: Trace plots for coefficients of dynamically fit models. LB0 refers to the endogenous learn-burn model, LBX refers to the full learn-burn model, and LTB refers to the full learn-turn-burn model.

3.5 Discussion

In this chapter, we have shown that the SRL has several worthwhile extensions. First, we showed how the SRL can be used in situations where the covariates are not all equally easy or costly to collect, and we wish to take this into consideration during the model selection process. The SRL can in many cases substantially decrease the cost of future data collection while also only minimally impacting the predictive quality of the model. Second, we showed how the SRL can be utilized in the time series regression framework to build quality forecasting models using both endogenous and exogenous components. The SRL in the time series regression setting is especially effective in the presence of complicated seasonality, where it is unclear which frequencies are important; the SRL was able to forecast the number of ER arrivals more successfully than other state-of-the-art methods. Finally, we applied the SRL in a new learn-turn-burn framework, where we found that the SRL could balance the tradeoff of model stability with model adaptivity quite well.

Though we have omitted many of our results for the sake of brevity, each of these extensions of the SRL are seeds of future work. We have envisioned many simulation studies and applications that could be carried out to further explore each of these methods, and we plan to fully flesh these out in later works.

3.6 Conclusion

The phenomenon of ranked sparsity, though quite abstract on its own, seems to appear quite frequently in applications. More often than not, investigators' zeal to leave no stone unturned in a scientific endeavour inevitably leads to the performance of many tests and the candidacy of many statistical models. If a primary hypothesis is not supported by the results of a study, perhaps there is some subgroup of the data that supports a slightly modified hypothesis. Or, perhaps a differently specified model yields new, more interesting insights. However, we know that this shift in the scientific question has serious statistical implications and impacts. Testing for all possible interactions will grossly inflate the number of parameters to choose from, and if unchecked will lead to overly opaque and misspecified models. Fortunately, our ranked sparsity methods can manage to account for this issue as well as many others.

REFERENCES

Akaike, H. (1974) A new look at the statistical model identification. *Selected Papers of Hirotugu Akaike* pp. 215–222. Springer.

Bien, J., Taylor, J. and Tibshirani, R. (2013) A lasso for hierarchical interactions. *The Annals of Statistics*, **41**, 1111–1141.

Boulesteix, A.-L., De Bin, R., Jiang, X. and Fuchs, M. (2017) IPF-lasso: Integrative-penalized regression with penalty factors for prediction based on multi-omics data. *Computational and Mathematical Methods in Medicine*, **2017**.

Breheny, P.J. (2018) Marginal false discovery rates for penalized regression models. *Biostatistics*, **20**, 299–314.

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics*, **5**, 232–253.

Chen, J. and Chen, Z. (2008) Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, **95**, 759–771.

Chipman, H. (1996) Bayesian variable selection with related predictors. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, **24**, 17–36.

Choi, N.H., Li, W. and Zhu, J. (2010) Variable selection with the strong heredity constraint and its oracle property. *Journal of the American Statistical Association*, **105**, 354–364.

Cryer, J.D. and Chan, K.-S. (2008) *Time Series Analysis with Applications in R*. Springer.

De Livera, A.M., Hyndman, R.J. and Snyder, R.D. (2011) Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, **106**, 1513–1527.

Fan, J. and Li, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, **96**, 1348–1360.

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

Friedman, J., Hastie, T. and Tibshirani, R. (2010) A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*.

Hao, N., Feng, Y. and Zhang, H.H. (2018) Model selection for high-dimensional quadratic regression via regularization. *Journal of the American Statistical Association*, **113**, 615–625.

Hoerl, A.E. and Kennard, R.W. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67.

Hurvich, C.M. and Tsai, C.-L. (1989) Regression and time series model selection in small samples. *Biometrika*, **76**, 297–307.

Hyndman, R.J. (2006) Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, **4**, 43–46.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., et al. (2018) *forecast: Forecasting Functions for Time Series and Linear Models*.

Hyndman, R.J. and Khandakar, Y. (2008) Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, **26**, 1–22.

Jones, S.S., Thomas, A., Evans, R.S., Welch, S.J., Haug, P.J. and Snow, G.L. (2008) Forecasting daily patient volumes in the emergency department. *Academic Emergency Medicine*, **15**, 159–170.

Lim, M. and Hastie, T. (2015) Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, **24**, 627–654.

Lumley, T. and Miller, A. (2017) *Leaps: Regression Subset Selection*.

Mallows, C.L. (1973) Some comments on Cp. *Technometrics*, **15**, 661–675.

McLeod, A. and Xu, C. (2018) *Bestglm: Best Subset Glm and Regression Utilities*.

Miller, R.E. and Breheny, P. (2019) Marginal false discovery rate control for likelihood-based penalized regression models. *Biometrical Journal*, to appear.

Mohebbi, M., Vanderkam, D., Kodysh, J., Schonberger, R., Choi, H. and Kumar, S. (2011) Google correlate whitepaper.

Peterson, R.A. (2017) *bestNormalize: A Suite of Normalizing Transformations*.

Peterson, R.A. and Cavanaugh, J. (2019) Ordered quantile normalization: A semiparametric transformation built for the cross-validation era. *Journal of Applied Statistics: Special Issue on Advances in Computational Data Analysis*, in revision.

Ravikumar, P., Lafferty, J., Liu, H. and Wasserman, L. (2009) Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**, 1009–1030.

Rissanen, J. (1978) Modeling by shortest data description. *Automatica*, **14**, 465–471.

Schwarz, G. (1978) Estimating the dimension of a model. *The Annals of Statistics*, **6**, 461–464.

Shedden, K., Taylor, J., Enkemann, S., Tsao, M., Yeatman, T., Gerald, W., et al. (2008) Gene expression-based survival prediction in lung adenocarcinoma: A multi-site, blinded validation study. *Nature Medicine*, **14**, 822–827.

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2013) A sparse-group lasso. *Journal of Computational and Graphical Statistics*, **22**, 231–245.

Simon, N. and Tibshirani, R. (2012) Standardization and the group lasso penalty. *Statistica Sinica*, **22**, 983.

Strawderman, R.L., Wells, M.T. and Schifano, E.D. (2013) Hierarchical bayes, maximum a posteriori estimators, and minimax concave penalized likelihood estimation. *Electronic Journal of Statistics*, **7**, 973–990.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, **58**, 267–288.

Wood, S. (2017) *Generalized Additive Models: An Introduction with R*, 2nd ed. Chapman; Hall/CRC.

Yuan, M. and Lin, Y. (2006) Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, **68**, 49–67.

Zeng, Y. and Breheny, P. (2017) The biglasso package: A memory- and computation-efficient solver for lasso model fitting with big data in R.

Zhang, C.-H. (2010) Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, **38**, 894–942.

Zou, H. (2006) The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**, 1418–1429.

Zou, H. and Hastie, T. (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, **67**, 301–320.