



University of Pennsylvania
ScholarlyCommons

Publicly Accessible Penn Dissertations


Summer 8-14-2009

Permuted Inclusion Criterion: A Variable Selection Technique

Shaun Lysen

University of Pennsylvania - Wharton School, slysen@gmail.com

Follow this and additional works at: <http://repository.upenn.edu/edissertations>

 Part of the [Applied Statistics Commons](#), [Multivariate Analysis Commons](#), [Statistical Methodology Commons](#), and the [Statistical Models Commons](#)

Recommended Citation

Lysen, Shaun, "Permuted Inclusion Criterion: A Variable Selection Technique" (2009). *Publicly Accessible Penn Dissertations*. 28.
<http://repository.upenn.edu/edissertations/28>

This paper is posted at ScholarlyCommons. <http://repository.upenn.edu/edissertations/28>
For more information, please contact libraryrepository@pobox.upenn.edu.

Permuted Inclusion Criterion: A Variable Selection Technique

Abstract

We introduce a new variable selection technique called the Permuted Inclusion Criterion (PIC) based on augmenting the predictor space X with a row-permuted version denoted X_{pi} . We adopt the linear regression setup with n observations on p variables. Thus, our augmented space has p real predictors and p permuted predictors. This has many desirable properties for variable selection. For example, this preserves relations between variables, e.g. squares and interactions and equates the moments and covariance structure of X and X_{pi} . More importantly, X_{pi} scales with the size of X . We motivate the idea with forward selection. The first time we select a predictor from X_{pi} , we stop. As this depends on the permutation, we simulate many times and create a distribution of models and stopping points. This has the added benefit of quantifying how certain we are about stopping. Variable selection typically penalizes each additional variable by a prespecified amount. Our method uses a data-adaptive penalty. We apply this method to simulated data and compare its predictive performance to other widely used criteria such as C_p , RIC, and the Lasso. Viewing PIC as a selection scheme for greedy algorithms, we extend the PIC to generalized linear regression (GLM) and classification and regression trees (CART).

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Statistics

First Advisor

Andreas Buja

Keywords

variable selection, linear regression, permutation, CART, model selection

Subject Categories

Applied Statistics | Multivariate Analysis | Statistical Methodology | Statistical Models

Permuted Inclusion Criterion: A Variable Selection Technique

Shaun Lysen

A Dissertation

in

Statistics

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2009

Andreas Buja
Supervisor of Dissertation

Eric Bradlow
Graduate Group Chairperson

Acknowledgments

I would like to express my deepest gratitude to my advisor Andreas Buja. His unique perspective on all areas of statistics, especially multivariate data, data visualization, and statistical computing has had a profound impact on my way of thinking about data. A big Thank You to Ed George and Abba Krieger for being on my Ph.D. committee and providing me with useful feedback. I would also like to thank my fellow peers over the past 4 years who have impacted my statistical thinking: Kenny, Kartik, Frank, Blake, Alex, Mingyuan, Michael, Mike, Oliver, Sathya, James, and others

ABSTRACT

Permuted Inclusion Criterion: A Variable Selection Technique

Shaun Lysen

Andreas Buja, Advisor

We introduce a new variable selection technique called the Permuted Inclusion Criterion (PIC) based on augmenting the predictor space \mathbf{X} with a row-permuted version denoted \mathbf{X}_π . We adopt the linear regression setup with n observations on p variables. Thus, our augmented space has p real predictors and p permuted predictors. This has many desirable properties for variable selection. For example, this preserves relations between variables, e.g. squares and interactions and equates the moments and covariance structure of \mathbf{X} and \mathbf{X}_π . More importantly, \mathbf{X}_π scales with the size of \mathbf{X} . We motivate the idea with forward selection. The first time we select a predictor from \mathbf{X}_π , we stop. As this depends on the permutation, we simulate many times and create a distribution of models and stopping points. This has the added benefit of quantifying how certain we are about stopping. Variable selection typically penalizes each additional variable by a prespecified amount. Our method uses a data-adaptive penalty. We apply this method to simulated data and compare its predictive performance to other widely used criteria such as C_p , RIC, and the Lasso. Viewing PIC as a selection scheme for greedy algorithms, we extend the PIC to generalized linear regression (GLM) and classification and regression trees (CART).

Contents

1	Introduction	1
1.1	The Need for Variable Selection	3
1.2	Variable Selection in Linear Regression	5
1.3	Building Candidate Models – Common Algorithms	6
1.3.1	All Subsets	6
1.3.2	Stepwise Methods	7
2	Selection Criteria – When To Stop?	11
2.1	The Need for a Penalty	11
2.1.1	Constant λ	13
2.1.2	λ as a function of p	17
2.1.3	λ as a function of p and k	21
2.2	Other Variable Selection Schemes	24
2.2.1	L_1 methods	24
2.2.2	Data Resampling Methods	30

2.2.3	Bayesian Methods	34
2.3	False Selection Rate	36
3	Permuted Inclusion Criterion	40
3.1	Augmenting the Data by Permutation	40
3.2	Forward Selection with the PIC	41
3.3	PIC applied to a single predictor	43
3.4	Permutation Framework	44
3.5	The Multivariate Case	45
3.6	PIC as an adaptive choice of λ	49
3.7	How to Select the Model	54
3.8	Choice of α	54
3.9	Relation to the False Selection Rate	57
4	Rotations vs. Permutations	59
4.1	When is Permutation Valid?	61
4.2	When is Rotation Valid?	62
5	Extending the PIC	63
5.1	Generalized Linear Models	63
5.2	PIC with GLM	64
6	Simulation Results	67
6.1	Simulation Setup	67

6.2	Simulations: $p > n$	79
7	Permuted Selection and Trees	81
7.1	CART	81
7.1.1	Traditional CART Stopping Criteria: Pruning	83
7.1.2	Permuted CART Stopping Criterion	84
7.1.3	How to Adjust and Select a Model	85
7.1.4	Example	86
7.1.5	Choice of α	91
7.2	Tree Extensions	92
8	Conclusions	95
A	Simulation Results	97

Chapter 1

Introduction

Variable selection has been of great interest to the statistical community for decades. It is one of the most fundamental problems in statistical applications and has received a great deal of theoretical input. Suppose we have n observations of a response variable y along with a potentially large number of predictor variables $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p$ and we seek a model relating the predictors to the response. Not every predictor should be included in the model and there are a few reasons why. First, a predictor may have no relation to the response and thus has no reason to be included in the model. Second, although a predictor may have a relation to the response, after we adjust for the variables already included in the model, this relation becomes negligible. This often happens when our predictors are highly correlated. Third, we desire simple models with as few predictors as possible. This has a philosophical underpinning with Occam's Razor, but also a statistical underpinning with the bias-variance tradeoff – simple models are less variable.

Identifying which subset of variables is “best” to build a model is precisely the variable selection problem. For this reason, variable selection is synonymously known as subset selection. More generally, subset selection is a special case of model selection where each model corresponds to a distinct subset of the variables. In all that follows, our primary goal with variable selection will be prediction and we will measure goodness-of-fit by squared error.

A fundamental problem with building a model is that we use the same data to both select the subset and to evaluate its error rate, introducing bias into the selection problem. As a result, the error rate will almost certainly be larger on a new data set. To correct for this, many selection procedures have been developed. We propose a new selection scheme called the Permuted Inclusion Criterion (PIC) that mitigates selection bias with a penalty that adapts to the dimensionality and correlation structure of the data set. The fundamental idea behind the PIC is the generation of a reference data set to serve as fake data that has no relation to the response variable. We augment the predictor space with this fake data so that we now have $2p$ variables. Suppose the real data has no signal to explain the response. Then in theory, we should have no preference whether we select a real predictor or a fake predictor. We will make all of this much more precise in the following chapters, but this fundamental concept of augmenting the dataset with nonsense predictors is essential. We will show this idea is much more general and can be used with any greedy algorithm that selects a “best” variable.

As variable selection is best understood and most often studied with linear regression,

we will motivate the PIC from this context. We will in later chapters, however, extend the idea to generalized linear regression (GLM) and classification and regression trees (CART).

This thesis is organized as follows. The remainder of Chapter 1 presents the need and reasons for variable selection and the most common ways to build candidate models. Chapter 2 provides an extensive overview of the most commonly used variable selection criteria. In Chapter 3 we introduce and describe in detail the Permuted Inclusion Criterion. Chapters 4 and 5 illustrate various extensions to the PIC including how under a slightly different assumption, we can consider rotated versions of the data set. We also extend it to Generalized Linear Models. In Chapter 6, we perform extensive simulations and compare the performance of the PIC to other common schemes. We also illustrate its use in the $p > n$ case. In Chapter 7, we extend the PIC and apply it to building Classification and Regression trees.

1.1 The Need for Variable Selection

We now consider a few reasons why variable selection is so important. Squared error decouples into the squared bias plus the variance. These quantities nearly always trade off against one another. When we include too few predictors in our model, we have underfit which results in high bias with low variance. On the other hand, if we include too many predictors, we have overfit which results in low bias with high variance. Delicately balancing the bias-variance tradeoff is at the heart of variable selection. For example,

suppose that we have a variable \mathbf{X}_j that has a weak relationship with the response. By including it in the model, we will decrease the bias. However, the addition of an extra variable increases the model's variance, and if this increase is larger than the decrease in bias, we would be better off excluding this variable. Much of the variable selection literature is devoted to balancing this tradeoff and developing good selection criteria to select an optimal number of predictors.

An additional goal is interpretability. The fewer predictors we include in our linear model, the more interpretable the model is. In many scientific studies, the relationship between inputs (predictors) and output (response) is paramount and this relationship is better understood with fewer predictors. We clearly do not want to omit any important variables, but we desire as simple of a model as possible.

Related to interpretability is a philosophical motivation from Occam's razor that seeks "the hypothesis with the fewest assumptions and fewest entities among competing hypotheses". In our case, all the models have the assumptions of normally distributed independent errors and a linear in β relationship between predictors and response. The fewest entities then correspond to the model with the fewest number of nonzero coefficients. All else equal, we seek the most parsimonious model.

Finally, in the increasingly common case where the number of predictors p is greater than the number of observations n , variable selection is a requirement. Including all p variables is impossible since we have at most n linearly independent variables. We have different subsets of variables giving the same predictions. We need some way to

intelligently choose which variables appear in our model.

1.2 Variable Selection in Linear Regression

As variable selection is most frequently studied in the linear regression case, we develop our ideas in this framework. Linear regression models are the most prevalent in applications and often serve as a good first approximation to nonlinear models. In all that follows, we will assume the data is centered so that we do not concern ourselves with an intercept¹.

Consider the traditional linear regression setup:

$$\mathbf{y} = \mu + \epsilon = \mathbf{X}\beta + \epsilon \quad \text{where} \quad \mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)^T \in \mathbb{R}^n$$

$$\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p) \in \mathbb{R}^{n \times p} \quad \epsilon \in \mathbb{R}^n \quad \text{with} \quad \epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$$

We adopt the residual sum of squares (RSS) as our error rate defined by:

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

Variable selection requires two phases. First, we need to build a pool of candidate models and second, we need to select the best model from this pool. In this chapter we discuss building candidate models, devoting all of Chapter 2 to the selection problem.

¹Equivalently, we could force the intercept into all models we build and not consider it as a predictor

1.3 Building Candidate Models – Common Algorithms

One of the primary problems of subset selection is that we have 2^p possible models to consider. For p of even moderate size, an exhaustive search is infeasible. For example, with $p = 30$, we have a billion potential models. Moreover, suppose we did generate all 2^p models. Unless p is small, this is generally a bad idea. The larger the pool of models we select from, the larger the selection bias. Consequently, we seek a balance between a computationally feasible strategy that also discovers “good” models. We next discuss the most common methods: All Subsets, Forward Selection, Backward Elimination, and Stepwise.

1.3.1 All Subsets

All Subsets seeks the best fitting model from all possible subsets of variables. Since we build all possible models, all subsets is guaranteed to contain the globally optimal model. No other model building strategy can guarantee this. When we have a small number of predictors, all subsets can be a good idea. For example, suppose we have 10 predictors and when looking at the best subsets of size 5, we always see the same 3 predictors appearing in the list. This gives strong evidence these three variables should be in the final model. Additionally, when we have a larger number of predictors, clever methods exist for speeding up the algorithm such as branch and bound techniques (Furnival and Wilson, 1974), which eliminate infeasible models from consideration. However, all subsets still quickly grows computationally infeasible and exposes the model to greater selection bias.

1.3.2 Stepwise Methods

Forward Selection and Backward Elimination are similar methods that sequentially add or delete one variable at a time, respectively. In Forward Selection, we start with the null model with no predictors. At each step, we add the variable most correlated with the current residual vector. We then orthogonalize the residual vector and all other predictor variables not yet selected with respect to the variable just added. The variable “most correlated with the current residual vector” is equivalent to the most significant predictor from software output. That is, we seek the variable with the smallest p-value or equivalently, largest t-statistic in absolute value. We will however, focus on the square of the t-statistic or the F-statistic for reasons that will be clear. We continue adding variables until we have added them all or we satisfy a stopping criterion. The most widely used stopping criterion is the F-to-enter. We prespecify the F-to-enter before building the model, and the first time the largest F-statistic is smaller than the F-to-enter, we stop.

Backward Elimination proceeds with all variables in the model and sequentially deletes the variable least correlated with the current residual vector. We continue until we have dropped them all or we have reached some stopping criterion. Analogous to forward selection, the most widely used stopping criterion is the F-to-delete. We prespecify the F-to-delete and the first time the smallest F-statistic is larger than the F-to-delete, we stop. In the case $p > n$, backward elimination fails because we have ambiguity about where to start. Any n linearly independent variables span the entire space and yield a residual sum of squares of 0.

Forward Selection and Backward elimination strike a sensible balance between computational feasibility and finding good models but they do have their drawbacks. They can select vastly different models. For example, we can create a data set where the first variable to be added with Forward Selection is also the first variable to be deleted with Backward Elimination. As a result, we often combine these two strategies into one algorithm called Stepwise regression. Stepwise regression alternates between adding the most significant variable and then deleting the least significant variable. One reason Stepwise can be better than Forward Selection or Backward Elimination alone is collinearity. That is, once a variable is added to the model, it may have made other variables already in the model much less significant because of this collinearity and they should be removed from the model. If we were strictly moving in a forward direction, this would not be possible. Put another way, while all three of these strategies are greedy algorithms, Stepwise is less greedy than Forward Selection or Backward Elimination alone. Since we both add and delete variables, Stepwise requires both an F-to-enter and an F-to-delete criteria. One natural question to ask is will this algorithm converge? With Forward Selection and Backward Elimination, it's clear when we stop. With Stepwise, we may not be able to add an additional variable, but once we delete the least significant variable, the F-statistics all change, meaning we may be able to add a variable now. A sufficient condition for convergence is that the F-to-delete is smaller than the F-to-enter.

An additional problem with all three of these algorithms is that there is no reason why the best subset of k variables has any relation to the best subset of $k + 1$ variables.

In fact, we can construct an example where the best subset of k variables has an empty intersection with the best $k + 1$ variables. The above stepwise procedures would require the best k variables to be a subset of the best $k + 1$ variables since we only consider the addition or deletion of a single variable at a time. We might consider adding or deleting 2 or 3 variables at a time, but this increases the computational complexity, which was one of the main reasons we decided on these methods in the first place. We can think of these stepwise methods as sensible quick and dirty ways to build candidate models, but to understand there is no reason to believe we will be anywhere near the global optimum.

For the rest of this paper, we will focus on Forward Selection as it is the clearest framework to describe our method, and works in the case where $p > n$. We now describe it in much more detail.

Forward Selection

Forward Selection starts out with the null model with no predictors selected. Then at each step, we select the variable most correlated with the current residual vector.

We have the standard regression setup with n observations of a response \mathbf{y} and p predictors arranged in a matrix \mathbf{X} . We introduce the following notation. Let $\mathcal{I} = \{1, 2, \dots, p\}$ and let \mathcal{A} be a subset $\mathcal{A} \subseteq \mathcal{I}$ and $\mathcal{A}^c = \mathcal{I} \setminus \mathcal{A}$. That is, \mathcal{I} is the index set of all the variables, and we define \mathcal{A} as the active index set of variables currently in the model, and \mathcal{A}^c is the index set of variables yet to enter the model. We define \mathcal{A}_j as the active index set for the first j variables to enter. Thus we have the relation $|\mathcal{A}_j| = j$. More precisely,

we let i_j be the index of the j^{th} variable to enter. Thus $\mathcal{A}_j = \{i_1, i_2, \dots, i_j\}$. Because we are working with Forward Selection where we do not delete variables we have the relation

$$\mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \dots \subseteq \mathcal{A}_p$$

Result: Forward Selection

Step 0. We initialize $\mathcal{A}_0 = \emptyset$, $\hat{\mu}_0 = 0$, $\mathbf{X}^0 = \mathbf{X}$

for $j = 1$ **to** $\min(p, n)$ **do**

1. $i_j = \operatorname{argmax}_{i \in \mathcal{A}_{j-1}^c} \left| \operatorname{cor}(y - \hat{\mu}_{j-1}, \mathbf{X}_i^{j-1}) \right|$
2. $\mathcal{A}_j = \mathcal{A}_{j-1} \cup i_j$
3. Let $H_j = \frac{1}{\mathbf{X}_{i_j}^{j-1T} \mathbf{X}_{i_j}^{j-1}} \mathbf{X}_{i_j}^{j-1} \mathbf{X}_{i_j}^{j-1T}$
4. $\hat{\mu}_j = \hat{\mu}_{j-1} + H_j (y - \hat{\mu}_{j-1})$
5. $\mathbf{X}^j = (\mathbf{I}_n - H_j) \mathbf{X}^{j-1}$

end

Algorithm 1: Forward Selection Algorithm: Step 1 selects the variable index most correlated with the current residual vector. Step 2 adds that index to the active set. Step 3 computes the projection matrix to orthogonalize. Step 4 updates the current model fit and Step 5 orthogonalizes all remaining variables not yet selected.

This algorithm gives a set of $\min(p, n) + 1$ models. From this set we need to choose our “best” model. The following chapter gives common selection techniques. The interested reader can find a nice, succinct overview of variable selection (George, 2000), or for a more detailed account see (Miller, 2002).

Chapter 2

Selection Criteria – When To Stop?

All of the variable selection algorithms proposed in the last chapter share one big question – When do we stop? We introduced the F-to-enter idea but without giving any idea to how large it should be. We now put this question in terms of the canonical variable selection problem.

2.1 The Need for a Penalty

There is an inherent problem with the RSS criterion. For a fixed number of k nonzero coefficients, this is a sensible criterion to compare models. However, the addition of a $(k+1)^{\text{st}}$ variable to the model will always have a smaller RSS than the model with k variables. Therefore, in order to select between models of different sizes we need to penalize the addition of a new variable. More specifically, consider the following

$$\beta_{k,\lambda}^* = \arg \min_{\beta} \frac{(\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)}{\sigma^2} + \lambda k$$

where k is the number of nonzero predictors and λ is the penalty. Each additional variable is penalized by the constant term λ . This framework encompasses many of the most common variable selection techniques with different choices of λ . Note that we divide the RSS by the noise variance. Using this criterion with Forward Selection, we add a variable if and only if

$$\frac{\text{RSS}_{k+1}}{\sigma^2} + \lambda(k+1) \leq \frac{\text{RSS}_k}{\sigma^2} + \lambda k$$

or equivalently if

$$\frac{\text{RSS}_k - \text{RSS}_{k+1}}{\sigma^2} \geq \lambda$$

Practically speaking, we never know the noise variance and thus must use an estimate $\hat{\sigma}^2$.

We then have

$$\frac{(\text{RSS}_k - \text{RSS}_{k+1}) / \sigma^2}{\hat{\sigma}^2 / \sigma^2}$$

which follows the form of an F statistic. Under the null hypothesis that all remaining variables have zero coefficients, this follows the F distribution on 1 and $n-k$ degrees of freedom. The estimate $\hat{\sigma}$ is typically taken as the Root Mean Square Error (RMSE) of the model with k variables. Therefore, we see that we add a variable to the model if its F -statistic is larger than λ . We have a direct correspondence between λ and the F -to-enter criterion from the previous chapter. We now consider several popular variable selection

schemes that can roughly be broken into 3 groups: constant λ , λ as a function of p , and λ as a function of both p and k .

2.1.1 Constant λ

Searching for Significance: $\lambda \approx 4$

We start with a method that does not have a strong theoretical basis, but remains ubiquitous in introductory regression courses. We term this “searching for significance.” We start with a null model and keep adding the variable with the smallest p-value until no variables have a p-value smaller than $\alpha = .05$. Using this cutoff, we add a variable if its t-statistic is above approximately 2 in absolute value. This corresponds to an F-statistic of 4. We mention it briefly to serve as an anchor for other methods, as we are certain it is a criterion you have encountered in the past.

Mallows’ C_p : $\lambda = 2$

We define C_p as

$$C_p = \frac{\text{RSS}_k}{\hat{\sigma}^2} - n + 2k$$

We see that this is exactly the variable selection formulation we have above with $\lambda = 2$ with the exception of subtracting n . Since the number of observations is the same for all models we consider, this does not affect the ranking of models by C_p . Mallows motivated this statistic as an unbiased estimate of the model error scaled by the noise variance:

$$\frac{1}{\sigma^2} \|\mathbf{X}\beta - \mathbf{X}\hat{\beta}\|^2 = \frac{1}{\sigma^2} \|\mu - \hat{\mu}\|^2$$

To directly show the bias-variance breakdown, we expand this (dropping the σ^2 for now)

$$\mathbf{E}\|\mu - \hat{\mu}\|^2 = \mathbf{E}\|\mu - \mathbf{E}(\hat{\mu}) + \mathbf{E}(\hat{\mu}) - \hat{\mu}\|^2 = \mathbf{E}\|\mu - \mathbf{E}(\hat{\mu})\|^2 + \mathbf{E}\|\mathbf{E}(\hat{\mu}) - \hat{\mu}\|^2$$

These two terms are precisely the squared bias and the variance, respectively. The variance is exactly equal to $k\sigma^2$. The squared bias can be estimated by $\text{RSS}_k - (n - k)\sigma^2$. Practically speaking we have to estimate σ^2 which we typically take as the residual variance from fitting the full model. Plugging this value in for σ^2 and dividing both sides by it, we derive C_p .

The bias we mention above is bias from omitting a variable. The other bias to be very aware of is selection bias. While C_p is an unbiased estimate of the model error¹, the moment we use it to choose among many models, it's no longer unbiased. Mallows took great care to warn against using C_p to select a best model.

“The discussion above does not lend any support to the practice of taking the lowest point on a C_p plot as defining a “best” subset of terms. The present author feels that the greatest value of the device is that it helps the statistician to examine some aspects of the structure of his data and helps him to recognize the ambiguities that confront him.”

Despite this, C_p is often used to select a best model, and moreover, attributed to him! See also (Mallows, 1995)

¹assuming σ^2 is known

AIC: $\lambda = 2$

Akaike (1973) formulated the Akaike Information Criterion (AIC) from an information theory perspective. Suppose that the data was truly generated from a density g . If we consider any other density f , the Kullback Leibler Divergence (KL) is defined as

$$\begin{aligned} KL(g||f) &= \int_{-\infty}^{\infty} g(x) \log \frac{g(x)}{f(x)} dx = \int_{-\infty}^{\infty} g(x) \log g(x) - \int_{-\infty}^{\infty} g(x) \log f(x) \\ &= \mathbf{E}_g \log g(x) - \mathbf{E}_g \log f(x) \end{aligned}$$

where \mathbf{E}_g denotes expectation with respect to g . We assume f is parameterized by $\theta \in \Theta$. Our goal is to choose θ to minimize the KL between g and f . Note that the first term $\mathbf{E}_g \log g(x)$ only depends on g and is common to all models. Thus minimizing the KL is equivalent to minimizing $-\mathbf{E}_g \log f(x)$ – the expected negative log likelihood of f under g . One key problem is that we do not know g . However, we have a sample from it and can form the sample mean

$$\frac{1}{n} \sum_{i=1}^n \log f(x_i)$$

In the traditional linear regression setup and assuming σ^2 is known, $\log f(x)$ is given by the $RSS/2\sigma^2$ with some constants that do not matter. By doing a Taylor series expansion to second order around the expectation of the minimizer, we arrive at

$$-\mathbf{E}_g \log f(x) \approx -\frac{RSS}{2\sigma^2} + k$$

where k is the dimensionality of θ . In our regression setup, θ is the β vector. Since multiplying by a positive scalar does not affect our optimal choice, we often see AIC as

double the above quantity.

$$-2 \log f(x) + 2k$$

Consequently, this gives a penalty of $\lambda = 2$. We note that AIC is much more general applying to any model with a parameterized likelihood. We only applied it to linear regression in this context. Hurvich and Tsai (1989) derive a bias-corrected AIC more appropriate for small sample sizes given by

$$AIC_C = AIC + \frac{2(p+1)(p+2)}{n-p-2}$$

BIC: $\lambda = \log(n)$

While BIC does not have constant λ it's often mentioned with AIC, and falls best within this group. Schwarz (1978) motivated the Bayesian Information Criterion (BIC) from a model consistency framework. That is, asymptotically we select the correct model with probability approaching one. For $n \geq 8$ it penalizes models more harshly than does AIC. BIC is derived by the fact that maximum likelihood estimators (MLE) are asymptotic limits of Bayes estimators for a certain class of priors. If we expand these estimators to a second term, we get the BIC penalty. BIC is asymptotically equivalent to selection by Bayes factors.

BIC is further justified from a coding theory perspective. Rissanen (1978) formulated linear regression from a compression perspective called the Minimum Description Length (MDL). Simply stated, we prefer the model that compresses the data the most. MDL consists of two steps. First, we select a model \mathcal{H} to use, and then, we communicate the

data \mathcal{D} under the model \mathcal{H} up to a prespecified precision (MacKay, 2003). Optimal codes communicate the string s in $\lceil -\log_2 p(s) \rceil$ bits, where p is the probability of s (Cover and Thomas, 2006). Ignoring rounding issues, this is precisely the negative log likelihood. In the standard linear regression setup, \mathcal{H} corresponds to which coefficients are nonzero. Rissanen postulated a precision of $\log(\sqrt{n})$ per parameter. Consequently, we have

$$-p(\mathcal{D}|\mathcal{H}) + |H| \log(\sqrt{n}) = -\frac{\text{RSS}}{2\sigma^2} + \frac{1}{2} \log(n) k$$

If we multiply this by 2, this is equivalent to BIC.

2.1.2 λ as a function of p

Suppose all p variables are normally distributed and orthogonal. That is

$$X_j^T X_k = 0 \quad \forall i \neq j$$

Additionally, we assume that we have a null model

$$y = \epsilon \quad \epsilon \sim N(0, \sigma^2)$$

where all of the variables have a zero coefficient. AIC and other constant penalties notoriously include too many irrelevant predictors (Breiman and Freedman, 1983). A desirable goal for our selection procedure would be to not overfit in this completely nonsense case. If we let F_i denote the F-statistic from including the i^{th} variable in the model, we correctly select the null model if

$$F_i \leq \lambda \quad \forall i$$

or equivalently,

$$\max_{i \in \{1, 2, \dots, p\}} F_i \leq \lambda$$

Our choice of whether we add a variable to the model depends on the distribution of the maximum of p (nearly) independent F statistics². This distribution can look vastly different from the F distribution. To quote Miller (2002), changing notation to match ours,

“The use of the terms ‘F-to-enter’ and ‘F-to-delete’ suggests that the ratios F_i have an F-distribution under the null hypothesis, i.e. that the model is the true model, and subject to the true residuals being independently, identically and normally distributed. This is not so. Suppose that after k variables have been entered, these conditions are satisfied. If the value of F_i is calculated for one of the variables chosen at random, then the distribution is the F-distribution. However, if we choose that variable which maximizes F_i , then the distribution is not an F-distribution or anything remotely like an F-distribution.”

Below we show precisely what that distribution looks like for various values of p . We fix the number of observations n at 100.

²Technically speaking, all F-statistics share the same denominator so they are not independent. Provided n is not too small, this effect should be negligible

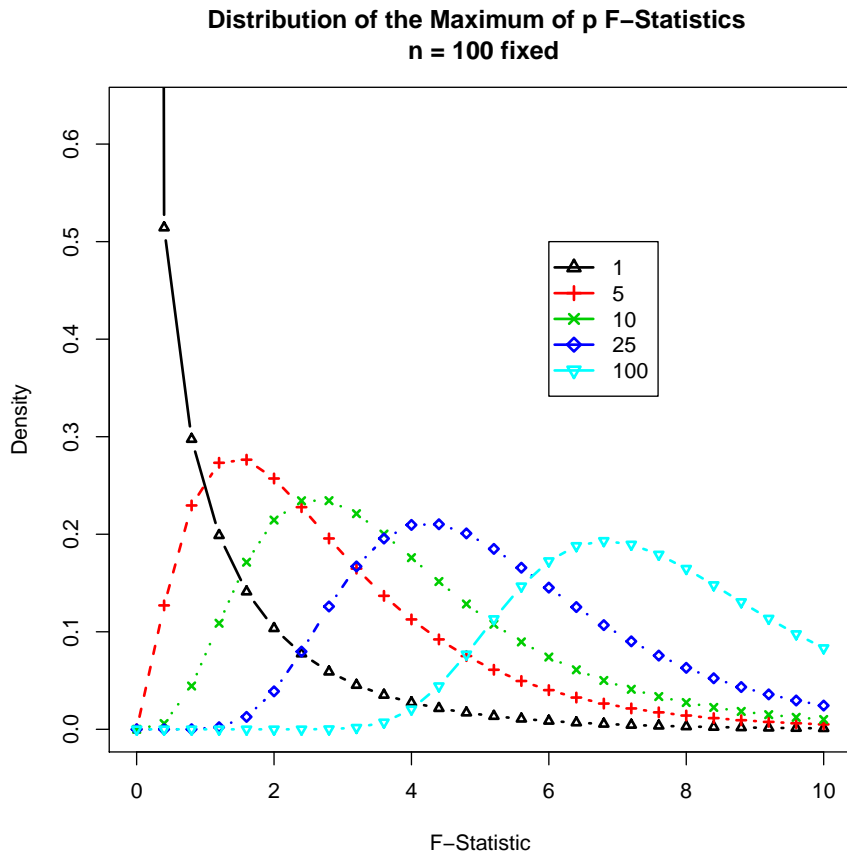


Figure 2.1: Distribution of the maximum F-statistic for p independent normal random variables with a response y that is also a normal random variable. This represents the maximum F to enter in a model that is pure noise.

The above figure clearly shows that our choice of λ should be increasing in p . This is intuitively clear since the larger our pool of variables to select from, the larger we expect this maximum to be. The leftmost distribution corresponds to $p = 1$. This is the F-distribution that all standard software packages would calculate p-values with respect to. While this plot intuitively motivates the idea that λ should be increasing in p , The

Risk Inflation Criterion makes it rigorous.

RIC: $\lambda = 2 \log(p)$

George & Foster motivate the Risk Inflation Criteria (Foster and George, 1994) by considering the worst possible inflation of risk due to selection, relative to the true model.

More specifically, define the Risk Inflation (RI) as

$$RI(\hat{\beta}) = \sup_{\beta} \frac{\mathbf{E}\|\mathbf{X}\beta - \mathbf{X}\hat{\beta}\|^2}{\mathbf{E}\|\mathbf{X}\beta - \mathbf{X}\hat{\beta}^*\|^2}$$

where we define $\hat{\beta}^*$ to be the estimated β vector if an oracle told us precisely which coefficients are nonzero. Choosing $\lambda = 2 \log(p)$ in the canonical variable selection problem yields a model that is minimax with respect to RI when the predictors are orthogonal. This penalty is important. It is the first one that grows with the size of the predictor space. As the graphs above demonstrate, this is a desirable trait. Additionally, the expected value of the maximum F -statistic is asymptotically $2 \log p$ motivating this penalty even more. This penalty also coincides with the universal threshold for wavelets developed independently by Donoho and Johnstone (1994). Another illuminating connection is between the RIC and the Bonferroni correction. When we are in a multiple testing framework with p tests and seek control over the Familywise Error Rate (FWER) at level α , we conduct each individual test at level α/p . If we translate the α/p quantile to the F -distribution, this is asymptotically sandwiched for large p between $2 \log p$ and $2 \log p - \log \log p$ (Foster and Stine, 2004).

2.1.3 λ as a function of p and k

Now suppose we remove the restriction of a null model so that the β vector might have nonzero entries and that we have added the first variable. Is it fair to penalize the next variable by the same $2 \log p$ penalty? If the remaining $p-1$ variables truly have zero coefficients, we are now selecting an F -statistic from one of two distributions. First, if we added the first variable correctly, then we are selecting the largest of $p - 1$ F -statistics. If instead we added the first variable incorrectly, then we are selecting the 2^{nd} largest of p F -statistics. Additionally, each of these distributions are truncated at the value of the maximum F -statistic. In either case, we are selecting from a distribution that is stochastically smaller than the largest F -statistic. The truncation only accentuates this fact. So, intuitively, we should relax the penalty. We illustrate this graphically below.

We now ask more generally, suppose we have added k variables, how should we penalize the addition of the $(k+1)^{\text{st}}$ variable.

False Discovery Rate

The penalty decreasing in k can be motivated from an important statistical perspective – the false discovery rate (FDR) (Benjamini and Hochberg, 1995). The false discovery rate is a multiple testing procedure that controls the proportion of falsely rejected hypotheses. In the variable selection context, a rejected hypothesis corresponds to adding a variable to the model. A falsely rejected hypothesis corresponds to adding a variable that we should not have. Consequently, each additional rejection, if its false, impacts the false discovery

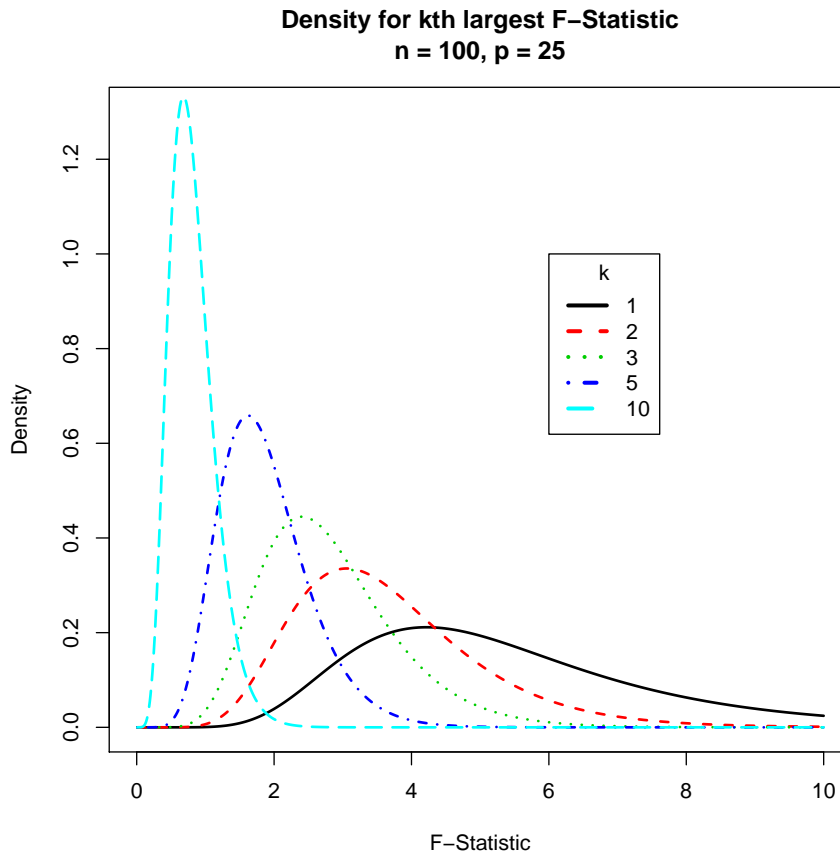


Figure 2.2: Distribution of various order statistics for the F-distribution for 100 observations of 25 independent normal random variables with a response y that is also a normal random variable. This motivates the idea why λ should be decreasing in k .

by a smaller amount since the total number of rejections – the denominator of the FDR – is increasing. For example, the addition of the first variable causes the FDR to be either 0 or 1 – a difference of 1; whereas the addition of the tenth variable causes the FDR to differ by $1/10$. Each variable added impacts the FDR less so we are more tolerant of an error. This corresponds to a λ that decreases in k .

Modern Methods: $\lambda = 2 \log(p/k)$

Many of the most recently developed variable selection penalties share this trait. The penalty λ should be increasing in p while decreasing in k as we add additional variables. Precisely, under the assumption that the number of nonzero predictors grows at a slower asymptotic rate than the number of predictors, i.e. $k = o(p)$, we have a family of penalties that are approximately $\lambda = 2 \log(p/k)$. For the first variable, the penalty is the same as RIC.

Foster and Stine (1999) derive the penalty $\lambda = 1/k \sum_{i=1}^k 2 \log(p/k)$ from information theory. Assuming $k = o(p)$, this is asymptotic to $2 \log(p/k)$.

Tibshirani and Knight introduced the Covariance Inflation Criterion (CIC) (Tibshirani and Knight, 1999b) which adjusts for overfitting by subtracting the average covariance between the predicted and actual response on permuted versions of the dataset. When the predictors are orthogonal, the penalty is $\lambda = 2/k \sum_{i=1}^k 2 \log(p/k)$. This penalty is twice as large as Foster and Stine's penalty.

George and Foster (2000) propose an Empirical Bayes approach where coefficients are drawn from the mixture prior $(1-w)\delta_0 + w N(0, C)$. δ_0 is a point mass at 0 representing a variable not in the model. They estimate the hyperparameters w and C from the data. They argue that this estimator penalizes the k th variable by a quantity close to $2 \log((p + 1 - k)/k)$.

Birge and Massart (2001) studied model selection under a class of penalties including $\lambda = 2 \log(p/k)$ and developed nonasymptotic risk bounds over l_p balls.

Abramovich et al. (2005) connect asymptotic minimaxity and multiple testing for a wide range of sparsity classes under a False Discovery Rate framework that penalizes models closely to $2 \log(p/k)$

2.2 Other Variable Selection Schemes

2.2.1 L_1 methods

All of the preceding methods can be viewed in another way: regularization of the β vector by the L_0 norm. That is we select β to minimize

$$\beta_{k,\lambda}^* = \arg \min_{\beta} \frac{(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)}{\sigma^2} + \lambda \|\beta\|_0$$

where $\|\beta\|_0 = \sum_{i=1}^p I(\beta_i \neq 0)$

As discussed above, one of the inherent difficulties of this problem is searching over all 2^p subsets which grows exponentially in p . One way to generalize this criterion is to consider a different norm. Suppose we replace the L_0 norm with a general L_q norm.

$$\beta_{q,\lambda}^* = \arg \min_{\beta} \frac{(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)}{\sigma^2} + \lambda \|\beta\|_q$$

This is exactly what is known as bridge regression (Frank and Friedman, 1993). Common specific cases correspond to $q = 2$: Ridge regression (Hoerl and Kennard, 1970) and $q = 1$: the Lasso (Tibshirani, 1996).

We can gain additional insight by considering a few base cases. Assume that the columns of \mathbf{X} have unit norm and are orthogonal – that is $\mathbf{X}^T \mathbf{X} = \mathbf{I}$. Then we can relate

common methods to the ordinary least squares estimates (OLS)

1. Ridge: $q=2$

$$\hat{\beta}_j^{Ridge} = (1 + \lambda)^{-1} \hat{\beta}_j^{OLS}$$

2. Lasso: $q=1$

$$\hat{\beta}_j^{Lasso} = \text{sgn}(\hat{\beta}_j^{OLS}) (|\hat{\beta}_j^{OLS}| - \frac{\lambda}{2})_+$$

3. Subset Selection (SS): $q=0$

$$\hat{\beta}_j^{SS} = \beta_j^{OLS} I(|\hat{\beta}_j^{OLS}| \geq \lambda)$$

Ridge shrinks the β vector by a multiplicative factor, but never setting any coefficients to 0. The Lasso performs soft thresholding, shrinking each coefficient towards zero by a constant amount. If this constant is greater than $\hat{\beta}_j$, it sets the coefficient to 0. Subset Selection performs hard thresholding, or the “keep or kill” strategy, by either leaving each coefficient unchanged or setting it equal to 0.

Lasso

One fact about Bridge regression is when $q \geq 1$, this penalized criterion performs shrinkage of the β vector as we see with the Lasso and Ridge penalties. The literature on the benefits of shrinkage is vast. See (Stein and James, 1961; Lehmann et al., 1998) for its origins in estimating the sample mean. For many shrinkage examples in modern statistics where shrinkage improves prediction, see (Hastie et al., 2001). An additional benefit is that the penalized criterion is convex. For subset selection $q = 0$, which is non-convex,

finding the “best” subset is NP-hard. Changing to a convex penalty, the solution is more easily found through widely available convex optimization software. For a good reference on convex optimization see Boyd and Vandenberghe (2004).

On the other hand, Bridge regression when $q \leq 1$, performs selection of the β vector, setting some coefficients equal to 0. The Lasso with $q = 1$ sits right at the boundary of these two operations. In fact, that is what Lasso stands for: Least Absolute Shrinkage and Selection Operator. The Lasso can be viewed as the closest convex approximation to the variable selection problem, replacing the L_0 norm with the L_1 norm. Additionally, the L_1 penalty is continuous so that we are able to see the entire profile of regression coefficients as we vary the penalty λ . At the two extremes, $\lambda = 0$ yields the full OLS model while $\lambda = \infty$ yields the null model. Figure 2.3 is an example. We also include the more common Lasso coefficient profile where the x-axis is the fraction relative to the full OLS fit.

Least Angle Regression

Least Angle Regression (LARS) (Efron et al., 2004) is one of the most fascinating recent developments in the linear model literature. LARS can be viewed as a smooth, continuous less-greedy version of Forward Selection. Forward Selection and LARS both start by selecting the variable that is most correlated with the response. Where they differ is how much they move in that direction. Forward Selection moves along that direction until the residual vector is orthogonal to it. If we instead imagine smoothly moving in that

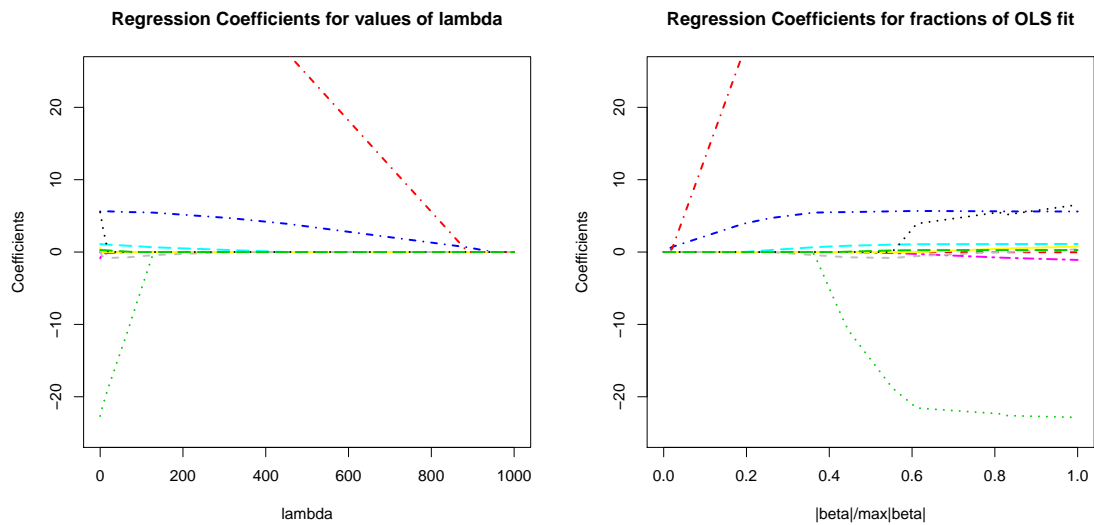


Figure 2.3: Coefficient profile for the Lasso as we vary λ on the left, and as we vary the fraction relative to the full OLS fit on the right. This is the diabetes data taken from Efron et. al (2004)

direction until we reach a point where another variable is equally correlated, we stop and change directions to the one that is equiangular to the two equally correlated variables. This is best illustrated with a geometric example.

We have a response variable y with two predictor variables \mathbf{X}_1 and \mathbf{X}_2 . The variable most correlated with y is equivalent to the variable which forms the smallest angle with it. In this case, we select \mathbf{X}_2 first. Both Forward Selection (FS) and LARS proceed in this direction. They differ by how far they travel.

Here we see the precise path that FS and LARS take. Forward Selection proceeds along \mathbf{X}_2 until the residual vector is orthogonal to it. It then moves in this orthogonal direction until it reaches y . In contrast, LARS proceeds along \mathbf{X}_2 until the point that the

Forward Selection vs. Least Angle Regression

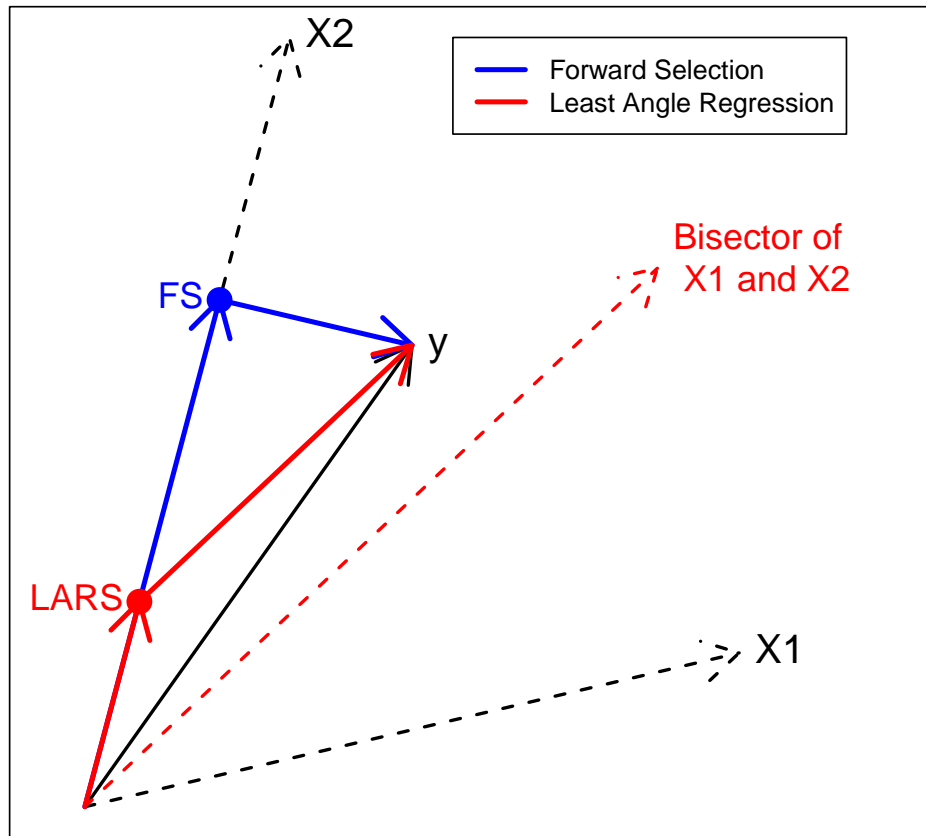


Figure 2.4: Both Forward Selection and Least Angle Regression select X_2 as the direction to move. Forward Selection travels until the residual vector is orthogonal. Least Angle Regression travels to the point where the residual vector is equally correlated with X_1 and X_2 and then moves along their angle bisector

residual vector is equally correlated – equivalently forms the same angle – with \mathbf{X}_1 and \mathbf{X}_2 . It then proceeds in the direction equiangular to \mathbf{X}_1 and \mathbf{X}_2 – the angle bisector. Unfortunately, we can only view this in two dimensions but the LARS algorithm generalizes to more than two directions. We add a new variable to the model each time its correlation with the current residual vector matches the correlation with those variables already in the model. We then recompute the equiangular direction with the new variable added, and proceed in that direction. LARS, just like Forward Selection, eventually reaches the Ordinary Least Squares (OLS) solution. It just does so in a less greedy manner. The geometric reasoning behind LARS is what gives it its name. At any given point in the algorithm, those variables that form the least angle with the current residual vector are included in the model. Any variables forming a larger angle are excluded.

Even more remarkable, with slight modifications of the algorithm, we can derive the entire path of Lasso solutions and the infinitesimal forward stagewise solutions. We point the interested reader to the original paper (Efron et al., 2004) and to a follow-up equating Lasso and Forward Stagewise in an expanded predictor space (Hastie et al., 2007).

Dantzig Selector

The Dantzig Selector (Candes and Tao, 2007) is a recent development especially suited to the $p > n$ case, where we can recover the nonzero components of the β vector with large probability under the uniform uncertainty principle on \mathbf{X} . More formally, the Dantzig

Selector is defined to be the solution to the convex problem

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_{l_1} \quad \text{subject to} \quad \|X^t r\|_{l_\infty} \leq \lambda \sigma$$

where $r = \mathbf{y} - \mathbf{X}\beta$ is the residual vector. They recommend $\lambda = \sqrt{2 \log p}$, which coincides with the RIC penalty. Candes and Tao showed that the program can be recast as a linear program, speeding up computation time. A remarkable result is that the mean squared error of β is within a logarithmic factor of the mean squared error if an oracle told us precisely which coefficients were nonzero. This result is not asymptotic. We should note the mean squared error above is for the β vector, not the $\mathbf{X}\beta$ vector for prediction. Efron et al. (2007) show the predictive performance of the Dantzig Selector relative to the Lasso to be similar in some cases and inferior in others. In the rejoinder, Candes and Tao specifically note the applications in biomedical imaging, genomics, and data conversion where estimating β is paramount.

2.2.2 Data Resampling Methods

Resampling methods attack variable selection from a different perspective by sampling repeatedly from the data to mimic what would happen with new data.

Cross Validation

The fundamental idea behind cross-validation (CV) is to divide the data into two parts and use the first part to build the model and the second part to evaluate the fit. We generally focus on K-fold cross-validation which divides the data set randomly into K equal (or

nearly equal) parts. We denote these parts by $\Gamma_1, \dots, \Gamma_K$ and let $\Gamma^{(k)}$ be the data with Γ_k deleted, $k = 1, \dots, K$. Suppose we have a sequence of forward selection models M_0, M_1, \dots, M_p with M_j the model with j variables. Then for each k , we carry out forward selection on $\Gamma^{(k)}$ generating models $M_0^k, M_1^k, \dots, M_p^k$ with respective sizes $0, 1, \dots, p$. Note that there is no reason for M_j^k to have the same j variables as M_j since Forward Selection may select different variables on different subsets. Then for each model size j we evaluate its cross-validated error (\widehat{CV}) as

$$\widehat{CV}(j) = \frac{1}{n} \sum_{k=1}^K \sum_{(y_i, \mathbf{x}_i) \in \Gamma_k} (y_i - \hat{\mu}_k(\mathbf{x}_i, M_j^k))^2$$

where $\hat{\mu}_k$ is the predictand evaluated on the left out subset Γ_k under model M_j^k .

We select model size $k = \arg \min_j \widehat{CV}(j)$. Stone (1977) showed that leave-one-out cross-validation ($K = n$) is asymptotically equivalent to AIC. However, Breiman and Spector (1992) recommend five-fold cross validation for variable selection based on simulation results because leave-one-out CV tends to select the same model as the entire data too often.

Both leave-one-out and five-fold CV are inconsistent for model selection. Shao (1993) showed that they tend to include too many variables. He proves we can ensure model consistency by letting the number of observations left out n_v satisfy $n_v/n \rightarrow 1$ as $n \rightarrow \infty$.

Bootstrap

The bootstrap attacks the variable selection problem by sampling with replacement from the empirical distribution. The bootstrap is typically applied in one of two ways. First, we can bootstrap the residuals. We start by fitting the full model estimating regression coefficients $\hat{\beta}$ and the residuals e_i , $i = 1, \dots, n$. We then studentize the residuals defined by $e_i^* = e_i / \sqrt{1 - h_i}$ where h_i is the i^{th} diagonal element of the hat matrix $H = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$. We then generate for each observation, a new $y_j^* = x_j \hat{\beta} + e_j^*$ where e_j^* is sampled with replacement from the studentized residuals. Crucial to bootstrapping residuals is that they have constant variance. This method is suitable if we treat \mathbf{X} as fixed.

Second, we can bootstrap by sampling with replacement from the observed $(\mathbf{x}_i, \mathbf{y}_i)$ pairs. This method is appealing if we are working in the random- \mathbf{X} case. It also makes no assumptions on the model, unlike the homoscedastic error assumption above. Consequently, we can view bootstrapping (\mathbf{x}, \mathbf{y}) pairs as “more nonparametric” than bootstrapping residuals. One problem that may arise in the $p > n$ case is each bootstrapped data set has rank on average about 63% as large as the original data set.

Shao (1996) showed that this bootstrap scheme is not consistent. If we bootstrap residuals, we can ensure consistency by increasing the variability of the residuals. If we bootstrap pairs, we can ensure consistency by constructing smaller bootstrap samples with size n_b with $n_b/n \rightarrow 0$ as $n \rightarrow \infty$. For a thorough treatment of the bootstrap see (Efron and Tibshirani, 1997).

Little Bootstrap

Breiman (1992) introduced the little bootstrap as an alternative to C_p that does not suffer from such severe selection bias. Suppose we fit a model with j coefficients denoted $\hat{\mu}_j$. The model error can be written as

$$ME(\hat{\mu}_j) = \|\mu - \hat{\mu}_j\|^2 = \text{RSS}_j - \text{RSS}_p + \|\mu - \hat{\mu}_p\|^2 - 2(\epsilon, \hat{\mu}_p - \hat{\mu}_j)$$

where the subscript p denotes the full model with all predictors included and (\cdot, \cdot) is the inner product. We also estimate the residual variance from the full model fit $\hat{\mu}_p$. The first two RSS terms are directly computed. We can estimate $\|\mu - \hat{\mu}_p\|^2$ as $p\hat{\sigma}^2$ assuming the full model has no bias. We need an estimate for this last term. Breiman proposes the little bootstrap to estimate this term. We generate bootstrapped responses as

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i + \tilde{\epsilon}_i \quad \text{where} \quad \tilde{\epsilon}_i \sim N(0, t^2\hat{\sigma}^2)$$

Note that we add the error term to the original response \mathbf{y} and not the predicted response $\hat{\mathbf{y}}$ like the residual bootstrap of the previous section. We then fit the model using forward selection to the $(\tilde{\mathbf{y}}_i, \mathbf{x}_i)$ pairs with j variables and all p variables as above. Call these fits $\tilde{\mu}_j$ and $\tilde{\mu}_p$. We then calculate

$$\frac{1}{t^2}(\tilde{\epsilon}, \tilde{\mu}_p - \tilde{\mu}_j)$$

Repeat this B times and average. Breiman showed that

$$\frac{1}{t^2}\mathbf{E}(\tilde{\epsilon}, \tilde{\mu}_p - \tilde{\mu}_j) \approx \mathbf{E}(\epsilon, \hat{\mu}_p - \hat{\mu}_j)$$

for small t . Consequently, we plug this estimate in for the final term in the model error

expression above.

$$\widehat{ME}(\hat{\mu}_j) = \text{RSS}_j - \text{RSS}_p + p\hat{\sigma}^2 - \frac{2}{B} \sum_{b=1}^B (\tilde{\epsilon}^b, \tilde{\mu}_p^b - \tilde{\mu}_j^b)$$

where the superscript b corresponds to a particular bootstrap sample. Based on simulations, Breiman recommends $t = 0.6$ and $B = 40$. The best model is then chosen with respect to this model error estimate.

2.2.3 Bayesian Methods

The Bayesian view on variable selection has seen a great deal of research in the past two decades. The fully Bayesian approach is as follows. Suppose that we have 2^p models denoted by M_1, M_2, \dots, M_{2^p} each of which corresponds to a distinct subset of the variables $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p\}$. We need two prior probabilities. First, we need a prior probability for the model M_γ which we denote $\pi(M_\gamma)$ and given the model, we need priors for each regression coefficient, denoted by $\pi(\beta_j|M_\gamma)$, $j = 1, 2, \dots, p$. The posterior probability for model M_γ is given by

$$\pi(M_\gamma|\mathbf{y}, \mathbf{X}) = \frac{p_\gamma(\mathbf{y}|\mathbf{X})\pi(M_\gamma)}{\sum_{k=1}^{2^p} p_k(\mathbf{y}|\mathbf{X})\pi(M_k)}$$

where

$$p_\gamma(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\beta_\gamma, \mathbf{X})\pi_\gamma(\beta_\gamma|M_\gamma)d\beta_\gamma$$

Selection occurs based on these posterior probabilities. The clear obstacle is then how do we choose the priors. Mitchell and Beauchamp (1988) introduced “spike and slab” priors, which for each predictor, consist of a mixture between a point mass at 0

(the spike) and a uniform distribution between $-c$ and c for some constant c (the slab). One obvious drawback is the finite support of the prior. One common alternative is to put normal priors on those variables appearing in the model. For the model itself, we put a binomial probability

$$P(M_\gamma) = w^k(1 - w)^{p-k}$$

where k is the number of variables in model M_γ . With this setup and fixing σ^2 , Chipman et al. (2001) showed that under different parameterizations, we can generate the variable selection problem for any penalty λ (e.g. AIC, BIC, RIC). Berger and Pericchi (1996) take a different approach, similar to cross-validation, by proposing to use part of the data to estimate the prior distributions and the remaining data to generate posterior probabilities.

Research prior to the advent of Markov Chain Monte Carlo (MCMC) methods focused on developing priors that minimally influence the posterior. The development of MCMC methods shifted the attention to fully specified prior distributions. The use of MCMC allows for much easier posterior calculations. Consequently, the primary obstacle now is how to intelligently search through the posterior. Markov Chain Monte Carlo model composition (MC^3) (Madigan et al., 1995; Raftery et al., 1997) proceeds similarly to Efroymson's stepwise algorithm. We start with a random subset of the variables. At each step, just like stepwise, we either add or delete a variable. The key difference is the choice of what variable to add or delete is stochastically guided. An alternative to MC^3 is Gibbs sampling. George and McCulloch (1993) modify the spike and slab prior by

allowing a mixture of two normal distributions, one with a very small variance. If the regression coefficient is sampled from this distribution, we can safely say its coefficient is 0 and should be left out of the model.

A related line of research in this context is Bayesian Model Averaging (Hoeting et al., 1999) where we take a weighted average of models sampled by the posterior. While this often gives better predictions, we lose our parsimonious goal of variable selection. Most or all variables will appear in the averaged model. For a general overview of Bayesian methods, see (Gelman et al., 2004).

2.3 False Selection Rate

We now discuss the the False Selection Rate (FSR) (Wu et al., 2007), a variable selection scheme most similar to the one we propose in the next chapter. We develop the FSR in detail in this section so that later we can contrast specific points with our method. We define the important variables as those for which $\beta_j \neq 0$ and unimportant variables as those for which $\beta_j = 0$. Ideally, we select all important variables and no unimportant variables. The FSR attempts to control the proportion of falsely selected unimportant variables. Suppose we specify an F-to-enter value that corresponds to significance level α and perform forward selection. The model selects $k(\alpha)$ variables of which $k_I(\alpha)$ are important and $k_U(\alpha)$ are unimportant. $k_U(\alpha) + k_I(\alpha) = k(\alpha)$. We of course do not observe k_U or k_I . We define the False Selection Rate (FSR) as the expected value of the proportion

of falsely selected variables, given by

$$\gamma(\alpha) = \mathbf{E} \left(\frac{k_U(\alpha)}{1 + k(\alpha)} \right)$$

We add 1 to the denominator to account for the intercept which is typically forced in the model and to avoid division by zero. In their paper, Wu, et al. consider two definitions, one the expectation of the ratio and another the ratio of expectations. We focus on the expectation of the ratio as it is the one they use in their simulations. The key goal is to select α_* so that $\gamma(\alpha_*) = \gamma_0$ for some prespecified false selection rate γ_0 . To ensure uniqueness, we take

$$\alpha_* = \sup\{\alpha : \gamma(\alpha) \leq \gamma_0\}$$

For example, $(k_I, k_U) = (3, 1)$ and $(k_I, k_U) = (7, 2)$ both give an FSR of 0.2. We prefer the larger model.

To control the FSR, they augment the predictor space with pseudo-variables that by design have no relation to the response. Consequently, by monitoring the number of pseudo-variables selected, and under the assumption that the pseudo-variables behave like the unimportant real predictors, we can estimate the FSR. We make our notation precise. Whenever we use p we mean the total number of predictors. If we attach a subscript to p , we mean the total number of predictors of that type, e.g. p_I is the total number of important predictors, p_U unimportant. We also introduce Z to be the pseudo-predictors. Consequently, we have a total of $p = p_I + p_U + p_Z$ predictors and at entry significance level we select a model size of $k(\alpha) = k_I(\alpha) + k_U(\alpha) + k_Z(\alpha)$. Ideally we want

to control

$$\gamma(\alpha) = \frac{k_U(\alpha)}{1 + k_U(\alpha) + k_I(\alpha)}$$

The denominator does not need to be estimated since we know how many total variables we selected $k(\alpha)$ and we know how many of those are pseudo-predictors $k_Z(\alpha)$, so the denominator is simply $1 + k(\alpha) - k_Z(\alpha)$. The numerator requires more effort. We need to make an assumption that on average the proportion of selected unimportant real predictors is equal to the proportion of selected pseudo-predictors for all α . That is,

$$\mathbf{E} \frac{k_U(\alpha)}{p_U} = \mathbf{E} \frac{k_Z(\alpha)}{p_Z}$$

and if we solve for $k_U(\alpha)$, we get

$$k_U(\alpha) = \frac{p_U}{p_Z} k_Z(\alpha)$$

Unfortunately, we also do not know p_U . So, we take an optimistic estimate and assume that among real predictors selected, we only selected important ones, i.e. $k_U(\alpha) = 0$. That is

$$\hat{p}_U = p_U + p_I - k_U(\alpha) - k_I(\alpha) = p - p_Z + k(\alpha) - k_Z(\alpha)$$

Every quantity on the right hand side is directly observable. In practice, we repeat this many times, generating new pseudo-variables and taking averages to estimate \hat{p}_U , $k(\alpha)$, and $k_Z(\alpha)$. Wu et al. use $B = 500$. We denote these averages as $\bar{\hat{p}}_U$, $\bar{k}(\alpha)$ and $\bar{k}_Z(\alpha)$, respectively.

We now have our estimate of the FSR

$$\hat{\gamma}(\alpha) = \frac{\bar{\hat{p}}_U \bar{k}_Z(\alpha) / p_Z}{1 + \bar{k}(\alpha) - \bar{k}_Z(\alpha)}$$

We then select

$$\alpha_{**} = \sup\{\alpha : \hat{\gamma}(\alpha) \leq \gamma_0\}$$

Note that we estimate α_{**} from the augmented space with pseudo-predictors and that it does not coincide with α_* estimated from the actual data. The final model is selected by running forward selection with a p-to-enter of α_{**} on the real data.

One key issue we postponed until now involves the pseudo-variables. Namely, how many of them do we include and how do we generate them? Wu et al. propose four different methods.

1. Generate independent normal variables
2. Randomly permute the rows of \mathbf{X}
3. Generate independent normal variables and orthogonalize with respect to \mathbf{X}
4. Randomly permute the rows of \mathbf{X} and orthogonalize with respect to \mathbf{X}

The last two methods are simply the first two methods adjusted to ensure that every pseudo-variable has zero correlation with every real predictor. The last two methods can not be used in the case $p > n$, and also suffer in the case $p > n/2$ since the rank of the pseudo-variables is at most $n - p$ which is smaller than the rank of \mathbf{X} . Methods 2 and 4 also restrict the number of pseudo-variables to be precisely the same as the number of real variables $p_Z = p_I + p_U$. In their simulations results, Wu et al. selected the fourth method based on simulations. We will have more to say about the generation of pseudo-variables later. Our method selects a variant of method 2.

Chapter 3

Permuted Inclusion Criterion

We now get into the heart of this thesis, describing the data augmentation procedure and how we apply our variable selection scheme. We will show that in the univariate case, our method coincides with Pitman's classic permutation test. We will then go into detail about how to adjust our predictors after each step of Forward Selection. In the last section of this chapter, we will compare and contrast our method with the False Selection Rate.

3.1 Augmenting the Data by Permutation

Our procedure begins by augmenting the predictor space \mathbf{X} , which we will call the actual or real data with a copy of \mathbf{X} , denoted \mathbf{X}_π , in which the rows have been randomly permuted. We will call \mathbf{X}_π the permuted data or the fake data. We now have an augmented predictor space $\tilde{\mathbf{X}} = (\mathbf{X} \mid \mathbf{X}_\pi)$ which consists of n observations on $2p$ variables. For each actual predictor \mathbf{X}_j , we have a corresponding permuted predictor $\mathbf{X}_{\pi j}$. Both versions have

the same marginal distributions since we only permuted the data. Additionally, if we let

$$\text{var}(\mathbf{X}) = \mathbf{S},$$

$$\text{var}(\mathbf{X}) = \text{var}(\mathbf{X}_\pi) = \mathbf{S}$$

The covariance structure is exactly the same, because the inner products are left unchanged. This is an extremely desirable property for variable selection because the distribution of test statistics depends on the correlation structure of the data. Furthermore, \mathbf{X}_π preserves transformations and interactions between other variables. For example, suppose $\mathbf{X}_3 = \mathbf{X}_2 \cdot \mathbf{X}_1$, then $\mathbf{X}_{\pi 3} = \mathbf{X}_{\pi 2} \cdot \mathbf{X}_{\pi 1}$. Lastly, a trivial yet important observation is that \mathbf{X}_π scales with the size of \mathbf{X} . Both have p variables. The larger the pool of predictor variables to select from, the larger the pool of fake predictors to penalize variable selection. This coincides with the intuition of RIC that the penalty λ should be increasing in p .

3.2 Forward Selection with the PIC

Forward Selection proceeds in a greedy fashion. We start out with the null model with no predictors selected. Then at each step, we select the variable most correlated with the current residual vector. Now suppose that instead of selecting the most correlated predictor from \mathbf{X} , we select the most correlated predictor from the augmented data $\tilde{\mathbf{X}}$. As long as we have not yet chosen a permuted predictor from \mathbf{X}_π , this procedure is equivalent to selecting from \mathbf{X} , since the order of variable entry is fixed conditional on observing \mathbf{X} . We propose a simple stopping criterion: as soon as we would select a predictor from \mathbf{X}_π , we stop. We call this the Permuted Inclusion Criterion (PIC). Clearly, this depends on

the realized permutation. Thus, we will simulate many permutations and create an entire distribution of model sizes to select our final model. Ideally, we would sample from all $n!$ permutations. However, this is prohibitively large for moderate n , so we simulate N times. This seemingly ad-hoc stopping criterion possesses many sensible properties.

Suppose we have the null hypothesis that \mathbf{y} is a complete noise model: $\mathbf{y} = \epsilon$. Now consider selecting from \mathbf{X} and from \mathbf{X}_π separately. For \mathbf{X} we select the variable with the largest absolute correlation from a pool of p predictors with covariance matrix \mathbf{S} that has no relation to the response under the null hypothesis. For \mathbf{X}_π we select the variable with the largest absolute correlation from a pool of p predictors with covariance matrix \mathbf{S} that has no relation to the response because we broke the relationship by permuting. They only differ in the reasons why they have no relationship with the response \mathbf{y} – one hypothetical under the null hypothesis, and one actual because we manipulated the data through permutation. If the largest absolute correlation from \mathbf{X} is larger than the largest absolute correlation from \mathbf{X}_π , we add that variable to the model. Otherwise, we stop and select the null model. Under the null hypothesis, we would expect the choice between \mathbf{X} and \mathbf{X}_π to be equally likely.

Before we delve into the details of how to adjust the variables and select a model, we draw an illuminating connection in the case of a single predictor.

3.3 PIC applied to a single predictor

Suppose we have a single predictor variable denoted by lowercase \mathbf{x} , and its permuted analog by \mathbf{x}_π . If we adopt the forward selection framework, we have two possible models

$$M_0 : \mathbf{y} = \epsilon \quad \text{or} \quad M_1 : \mathbf{x}\beta + \epsilon$$

For a given permutation π , we select M_0 if $|\text{cor}(\mathbf{y}, \mathbf{x}_\pi)| > |\text{cor}(\mathbf{y}, \mathbf{x})|$. Otherwise, we select M_1 . Suppose we now take N total permutations and let π_j denote the j^{th} permutation. We sample π_j from the universe of all $n!$ permutations, which we denote $\mathbf{\Pi}^n$. Let C_0 be the count for the number of times we select M_0 . We have the following algorithm.

Result: Permuted Selection with a Single Predictor

We initialize N total permutations, and count variable $C_0 = 0$

```
for  $i = 1$  to  $N$  do  
  | Sample  $\pi_j$  from  $\mathbf{\Pi}^n$   
  | if  $|\text{cor}(\mathbf{y}, \mathbf{x}_{\pi_j})| \geq |\text{cor}(\mathbf{y}, \mathbf{x})|$  then  
  |   | Select  $M_0$   
  |   |  $C_0 \leftarrow C_0 + 1$   
  | else  
  |   | Select  $M_1$   
  | end  
end
```

Algorithm 2: Permuted Inclusion Criterion: Single Predictor Variable. We augment the predictor space with \mathbf{x}_{π_j} and we count the number of times that it has a larger absolute correlation than the actual data

This looks strikingly familiar to the permutation test for correlation between \mathbf{x} and \mathbf{y} (Pitman, 1937). In fact, let

$$\hat{P} = \frac{C_0 + 1}{N + 1}$$

then \hat{P} is the P-value for the test of correlation between \mathbf{x} and \mathbf{y} . This P-value is exact up to simulation error and does not depend on the error distribution. The only fact we need is independent data. We add one to the numerator and denominator because the observed correlation is typically included in the reference set. Under the null hypothesis, C_0 is uniformly distributed on the integers $\{0, 1, \dots, N\}$. This connection should not be surprising since adding a variable to the model is equivalent to testing whether its slope is 0, and testing whether a slope is 0 is equivalent to testing whether its correlation is 0. We often view a permutation test for correlation by considering many realizations of \mathbf{x}_π alone, not augmented with \mathbf{x} . However, we see that this alternative framework of augmenting the data and then selecting is equivalent for the single predictor case. Before we extend the PIC to the multivariate case, we develop some notation.

3.4 Permutation Framework

We define $\mathbf{\Pi}$ to be the space of all row permutations. Note

$$|\mathbf{\Pi}| = n!$$

We suppress the dependence on n since we typically view the number of observations as constant. Let $\pi \in \mathbf{\Pi}$ be an element of this space. Thus \mathbf{X}_π is a realized row-permutation

of \mathbf{X} .

Different realizations of \mathbf{X}_π will give different stopping points for Forward Selection. We define Π_j to be the subset of Π such that we have not stopped after j steps. That is, we have yet to select a variable from \mathbf{X}_π after j variables have been selected. For example, Π_2 corresponds to those permutations where the first two variables selected came from \mathbf{X} . Since we stop the first time we select from \mathbf{X}_π , we have the relation that

$$\Pi = \Pi_0 \supseteq \Pi_1 \supseteq \Pi_2 \dots \supseteq \Pi_p \supseteq \Pi_{p+1} = \emptyset$$

Clearly, Π_0 consists of all the permutations since we have not selected any variables yet. Also, since we have p real variables to select from, Π_{p+1} corresponds to the empty set.

3.5 The Multivariate Case

Until now, we avoided an important issue with forward selection. Recall that in traditional forward selection, we adjust not only the current residual vector but also all other predictors yet to enter the model. We now examine how to adjust the augmented space $\tilde{\mathbf{X}}$ and how it impacts the selection of later variables. We adopt notation from chapter 1. Let \mathbf{X}^j and \mathbf{X}_π^j denote the real and permuted spaces after we adjust for the first j variables entered in FS.

We have three desired goals with PIC.

1. At each step in the algorithm, we want $\text{var}(\mathbf{X}^j) = \text{var}(\mathbf{X}_\pi^j) \quad \forall j$

2. Assuming \mathbf{X}^j has no signal to explain the response, the choice between \mathbf{X}^j and \mathbf{X}_π^j is equally likely
3. At step j , \mathbf{X}_π^j corresponds to a permutation $\pi \in \Pi_j$

We mention these three goals because unfortunately, we will only be able to simultaneously satisfy at most 2 of these goals for different adjustment schemes. We desire the first goal because one of the main motivations for the PIC over traditional methods is that it adapts to the covariance structure of \mathbf{X} . This means the F statistics have the same correlations throughout all steps of forward selection. Traditional selection criteria mentioned in chapter 2 do not address correlated test statistics. We already mentioned the benefit of augmenting with permuted data is that $\text{var}(\mathbf{X}) = \text{var}(\mathbf{X}_\pi)$. This is true at the start of the algorithm. However, as we proceed with Forward Selection this may not continue to hold. It depends on how we adjust.

We desire the second goal because under the assumption that \mathbf{X}^j possesses no more signal to explain the current residual vector, we should be indifferent between \mathbf{X} and \mathbf{X}_π .

We desire the third goal because we want to be sampling from the right subset. For example, why would we consider adding a 4th variable if we already stopped after the 2nd variable. This goal essentially prevents us from re-permuting at each step of Forward Selection.

We adopt the notation $\mathbf{X}_{2:1}$ to mean \mathbf{X}_2 adjusted for \mathbf{X}_1 . In regression terms, this means we regress \mathbf{X}_2 on \mathbf{X}_1 and take the residuals. This is the part of \mathbf{X}_2 that is orthogonal to \mathbf{X}_1 . Without loss of generality, suppose that the columns of \mathbf{X} are ordered as they

would enter in Forward Selection, e.g. \mathbf{X}_1 is the first variable to enter, \mathbf{X}_2 the second, etc. Suppose that we add \mathbf{X}_1 to the model. The pertinent question is how do we adjust \mathbf{X}_π ? We propose three methods and discuss their relative merits.

The most natural choice would be to adjust all predictors $\widetilde{\mathbf{X}}$ with respect to \mathbf{X}_1 . This is how common software routines would implement Forward Selection if we just augmented the predictor space with \mathbf{X}_π . Unfortunately, we then lose our first goal. We do not have equal covariances. While this effect might be small after the first variable is selected, it becomes much more dramatic as we add more variables. One clear way to see the covariance structure is not preserved is by considering \mathbf{X}_1 vs. $\mathbf{X}_{\pi 1}$. After adjustment, $\mathbf{X}_{1:1} = \vec{0}$ clearly. However, $\mathbf{X}_{\pi 1} \neq \vec{0}$ unless π is the identity mapping. Consequently, \mathbf{X}^1 now has $p - 1$ nonzero columns while \mathbf{X}_π^1 has p nonzero columns. This adjustment scheme has the additional downside that after k steps \mathbf{X}^k has rank $p - k$ while \mathbf{X}_π^k has rank p , unfairly biasing towards \mathbf{X}_π in the case of no signal – a violation of our second goal.

More rigorously, we define the rank-1 matrix $H_1 = \frac{1}{\mathbf{X}_1^T \mathbf{X}_1} \mathbf{X}_1 \mathbf{X}_1^T$. This is the projection matrix into the direction of \mathbf{X}_1 . Consequently, the matrix $I_n - H_1$ is the adjustment matrix, where I_n is the $n \times n$ identity matrix. Assume that the columns of \mathbf{X} are centered. Then,

$$\text{Var}((I_n - H_1) \mathbf{X}) = \mathbf{X}^T (I_n - H_1)^T (I_n - H_1) \mathbf{X} = \mathbf{X}^T (I_n - H) \mathbf{X} = \mathbf{X}^T \mathbf{X} - \mathbf{X}^T H_1 \mathbf{X}$$

because $(I_n - H_1)$ is symmetric and idempotent. Similarly for \mathbf{X}_π ,

$$\text{Var}((I_n - H_1) \mathbf{X}_\pi) = \mathbf{X}_\pi^T (I_n - H_1)^T (I_n - H_1) \mathbf{X}_\pi = \mathbf{X}_\pi^T (I_n - H_1)^T \mathbf{X}_\pi = \mathbf{X}_\pi^T \mathbf{X}_\pi - \mathbf{X}_\pi^T H_1 \mathbf{X}_\pi$$

Note that the first terms in each of these are equivalent $\mathbf{X}^T \mathbf{X} = \mathbf{X}_\pi^T \mathbf{X}_\pi$ since inner products are preserved under row permutations, but it's the second term where they differ.

Specifically if we expand and factor $\mathbf{X}^T H_1 \mathbf{X}$, we have

$$\frac{1}{\mathbf{x}_1^T \mathbf{x}_1} (\mathbf{x}_1^T \mathbf{X})^T (\mathbf{x}_1^T \mathbf{X})$$

Note that $\mathbf{x}_1^T \mathbf{X}$ is a linear combination of the rows of \mathbf{X} with weights given by elements of \mathbf{x}_1 . For the second equation, we substitute \mathbf{X}_π wherever \mathbf{X} was

$$\frac{1}{\mathbf{x}_1^T \mathbf{x}_1} (\mathbf{x}_1^T \mathbf{X}_\pi)^T (\mathbf{x}_1^T \mathbf{X}_\pi)$$

and we see we have the same linear combination of rows, only for \mathbf{X}_π . Since the rows of \mathbf{X}_π do not coincide with the rows of \mathbf{X} this second term is not equal. We lose our goal of equal covariances, which is one of the primary motivations for the PIC.

We propose two ways to fix this. After we adjust \mathbf{X} for \mathbf{X}_1 , we could re-permute the adjusted data matrix \mathbf{X}^1 . Sample a new permutation $\pi' \in \mathbf{\Pi}$ to give a new \mathbf{X}_π^1 , which will share an identical covariance matrix with the adjusted \mathbf{X}^1 . We preserve the first goal mentioned above. Unfortunately, we violate the third goal. At step j we will be sampling π' from the entire universe of permutations $\mathbf{\Pi}$ and not from the subset $\mathbf{\Pi}_j$. Why would we base our decision to add a new variable at step j from a subset $\mathbf{\Pi} \setminus \mathbf{\Pi}_j$ that has zero probability of the model reaching that point? We instead take a different route.

The alternative way to preserve equal covariance matrices across each step is to adjust \mathbf{X}_π differently than \mathbf{X} . Rather than adjust \mathbf{X}_π with respect to \mathbf{X}_1 , we will adjust it with respect to \mathbf{X}_{π_1} . Adjusting \mathbf{X}_π this way ensures the covariance matrices are identical across all steps of Forward Selection. Additionally, because $\mathbf{X}_{\pi_1:\pi_1}$ is the zero vector, we preserve the number of nonzero variables. At step k both \mathbf{X}^k and \mathbf{X}_π^k will have $p - k$ nonzero

variables. This is extremely important. As k increases, the number of nonzero predictors of \mathbf{X}_π^k decreases. Since \mathbf{X}_π^k functions as a reference set to penalize Forward Selection, on average, the penalty is decreasing in k . However, with this adjustment scheme we unfortunately lose the second goal. Because the response variable \mathbf{y} and \mathbf{X} are adjusted with respect to \mathbf{X}_1 , they will lie in the same $(n - 1)$ -dimensional space. \mathbf{X}_π is adjusted with respect to $\mathbf{X}_{\pi 1}$ and consequently will not be orthogonal to \mathbf{X}_1 – the residual space. This means that even if \mathbf{X}^1 has no relation to the current residual vector, the probability of selecting from \mathbf{X}^1 will be greater than the probability of selecting from \mathbf{X}_π^1 . Fortunately, unless the sample size is small, this will have a minimal effect. Therefore, this is the adjustment scheme that we adopt for the PIC.

3.6 PIC as an adaptive choice of λ

The PIC differs from traditional variable selection methods where the F-statistic for the variable to enter must be above some pre-specified threshold, e.g. $\lambda = 2$ for Mallows' C_p . In contrast, the PIC yields a data-dependent choice of λ that adapts to the correlation structure and dimensionality of \mathbf{X} . Recall from Chapter 2 our criterion for adding a $(k+1)$ st variable:

$$\frac{(\text{RSS}_k - \text{RSS}_{k+1}) / \sigma^2}{\hat{\sigma}^2 / \sigma^2} \geq \lambda$$

Simply stated, we add a variable if its F -statistic is larger than λ . Let F_{k+1}^* denote the largest F statistic selected from \mathbf{X}^k , the predictor space adjusted for the first k variables. Similarly, let $F_{\pi_{k+1}}^*$ denote the largest F statistic selected from \mathbf{X}_π^k , the permuted space

adjusted for the first k variables. For a given simulation for the PIC, we add the $(k+1)^{\text{st}}$ variable if $F_{k+1}^* \geq F_{\pi_{k+1}}^*$. Consequently, $\lambda = F_{\pi_{k+1}}^*$.

Suppose we have the complete noise model $\mathbf{y} = \epsilon$ and the predictors are orthogonal. Then the PIC asymptotically coincides with the RIC penalty of $2 \log p$. Our result, however, is not asymptotic and we have finer control over what quantile of this maximal distribution to use as a cutoff. We illustrate this below by showing which quantile the RIC cutoff corresponds for various values of n and p .

We now compare how both the RIC ($\lambda = 2 \log p$) and what we term the modified RIC ($\lambda = 2 \log(p/k)$) penalty relate to the .95 quantile for more than just the first variable to enter. We simulated 1000 datasets of complete noise and performed Forward Selection on each of them, recording what the maximum F -statistic was at each stage. This mimics how large our data-dependent λ would be calculated from \mathbf{X}_π . Each dataset consisted of 100 observations on 25 variables and we considered two correlation schemes. First, we considered uncorrelated data, which we denote $\rho = 0$. This data corresponds to the red circles below. Second, we simulated data with an autoregressive correlation structure where the correlation matrix has entries $\rho_{i,j} = \rho^{|i-j|}$. We took $\rho = 0.9$. This data corresponds to the blue triangles. RIC is the dashed horizontal cyan line. Modified RIC is the dotted pink line. Modified RIC tracks the uncorrelated data remarkably well.

We note that these curves do not indicate how the various λ penalties will perform on prediction at all. We were curious how they related to various quantiles of the ordered F -distributions. The first two plots show the theoretical maximum F distribution in the

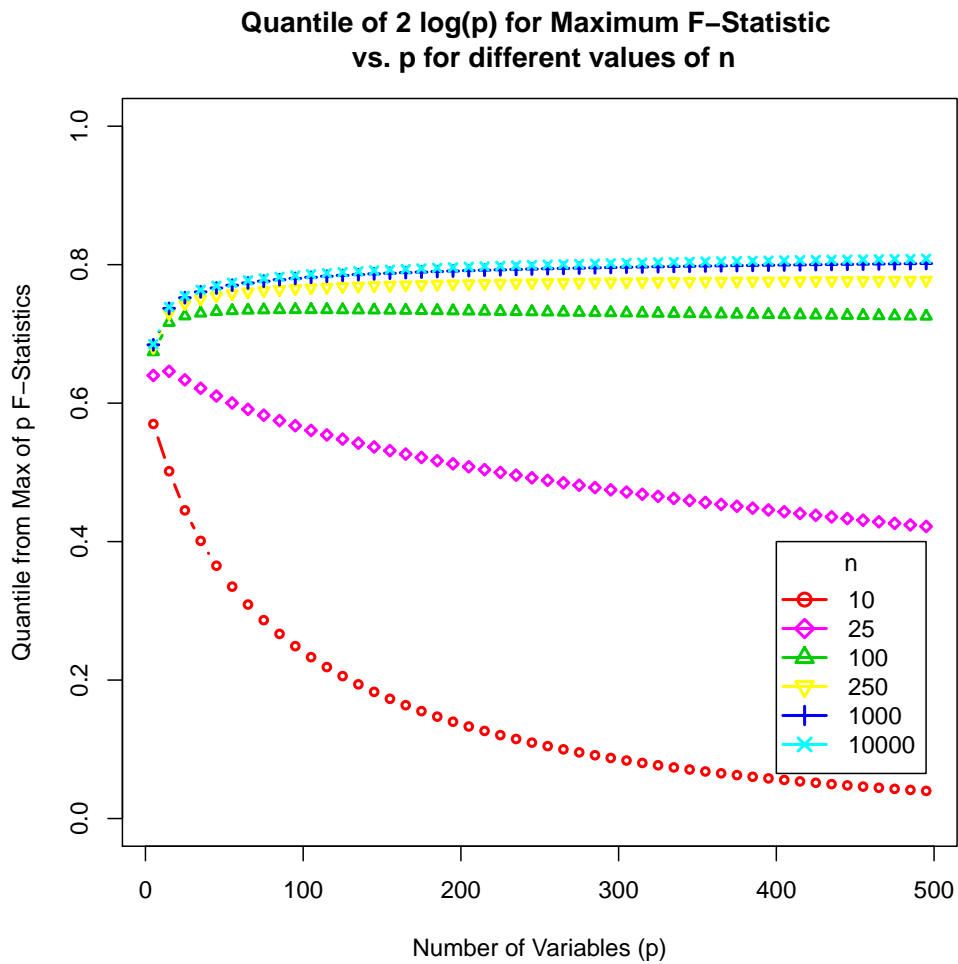


Figure 3.1: This shows the quantile corresponding to $2 \log(p)$ for the maximum F -statistic as p increases for different levels of n . First, we note that clearly n has a big impact for small sample sizes. Second, for moderate sample sizes, the curves are approaching a value around 0.8. This corresponds to a significance level of 0.20 for adding the first variable.

idealized case of orthogonal predictors. Even if the data is truly generated from an uncorrelated distribution, the sample will be correlated. That is why we simulated the λ cutoff

Quantile of $2 \log(p)$ for Maximum F-Statistic vs. n for different values of p

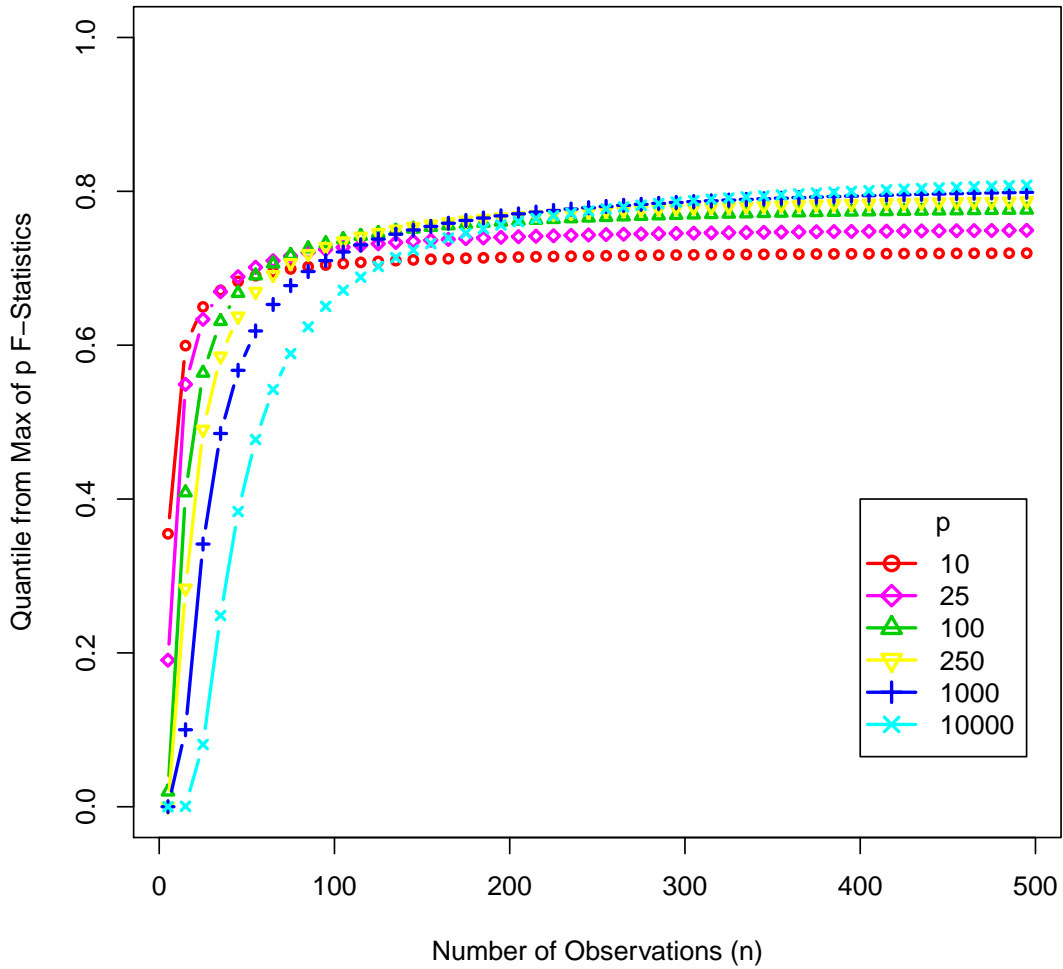


Figure 3.2: This shows the quantile corresponding to $2 \log(p)$ for the maximum F -statistic as n increases for different levels of p . We note the steep slope up until about $n = 100$. Again we have to be careful of small sample sizes.

for different values of k and to see how correlation impacts λ . We next discuss how to select our model.

.95 Quantile for Order Statistics of the F-Distribution
n = 100, p = 25

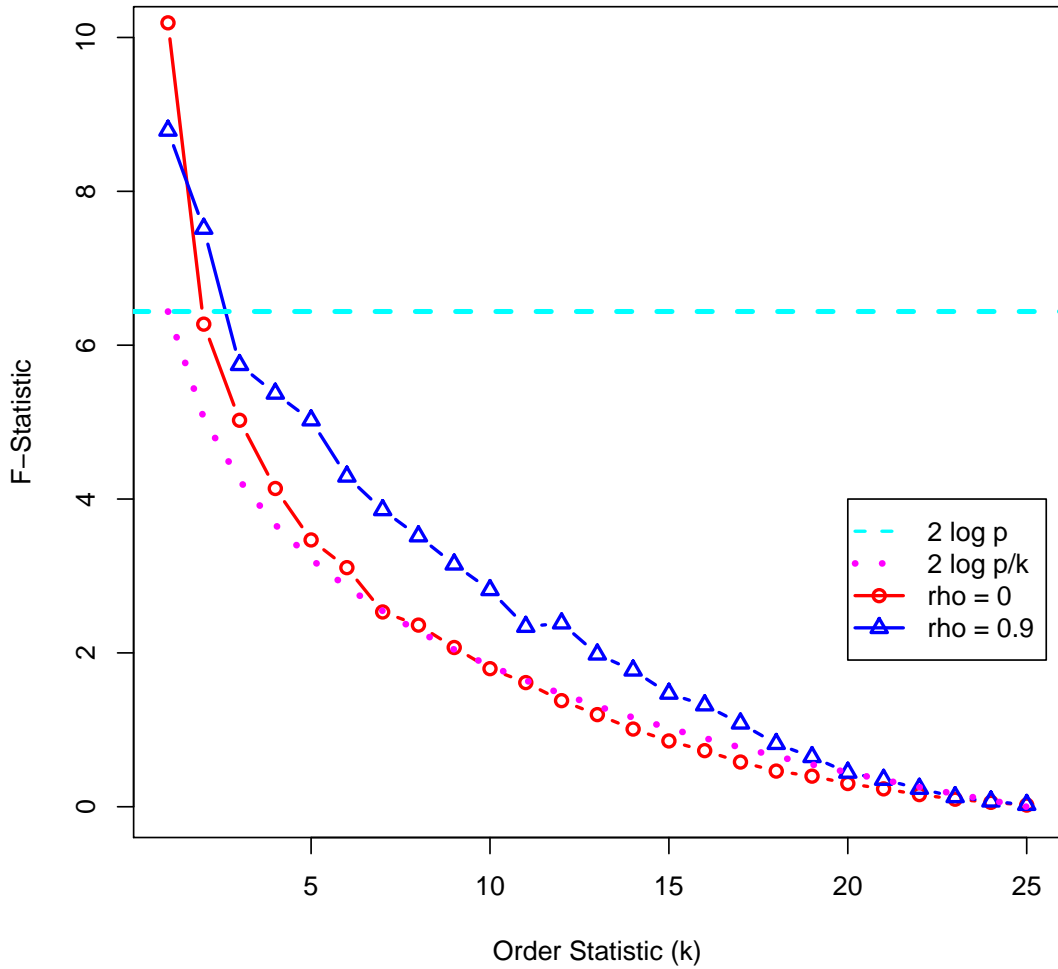


Figure 3.3: This plot shows the 0.95 quantile for λ as k increases. This is based on simulated data for two different correlation schemes. We see that clearly RIC is too harsh of a penalty as we add more variables. The modified RIC curve tracks the $\rho = 0$ curve extremely well, while the correlated $\rho = 0.9$ curve is consistently above the penalty. Note

3.7 How to Select the Model

We have discussed when to stop Forward Selection for a given permutation but we simulate this many times, creating a distribution of stopping points, and have not yet discussed how we select our final model. We propose a simple criterion: Specify a proportion α . The selected model size is chosen as the last point we have at least a $1 - \alpha$ proportion of our models. More rigorously, let N be the total number of simulations and let N_k denote the number of simulations that added at least k variables. Then the model size we choose is

$$k^* = \sup_k \{k : \frac{N_k}{N} \geq 1 - \alpha\}$$

Clearly, the choice of α is important. The smaller it is, the more parsimonious the model is.

3.8 Choice of α

We now investigate how to choose α based on simulations. This is the same setup for a set of simulations we will use later. Every model has 21 predictors generated from a multivariate normal distribution with mean 0 and autoregressive covariance structure

$$\text{cov}(x_i, x_j) = \rho^{|i-j|}$$

We consider $\rho = 0$ and $\rho = 0.7$. We center all predictor variables as well as the response so that we do not have to worry about an intercept. The nonzero coefficients are clustered

around \mathbf{X}_7 and \mathbf{X}_{14} with values given by

$$\beta_{7+j} = (h - j)^2 \quad \text{and} \quad \beta_{14+j} = (h - j)^2 \quad \text{for} \quad |j| < h$$

We consider $h = 0, 1, 2, 3,$ and 4 and denote these models $H_0, H_1, H_2, H_3,$ and H_4 . These correspond to models with $0, 2, 6, 10,$ and 14 nonzero coefficients, respectively. For each model we scale the coefficients so that the theoretical R^2 is 0.75 where

$$R^2 = \frac{(\mathbf{X}\beta)^T(\mathbf{X}\beta)}{(\mathbf{X}\beta)^T(\mathbf{X}\beta) + n\sigma^2}$$

We also consider a varying number of observations with $n = 50, 150,$ and 500 .

We simulate each of these 30 models 50 times and take the average of the model error defined as

$$ME(\hat{\beta}) = \|\mathbf{X}\beta - \mathbf{X}\hat{\beta}\|^2$$

We calculate this for $\alpha = .05, .10, .15, \dots .95,$ and plot the results for each of the 5 models.

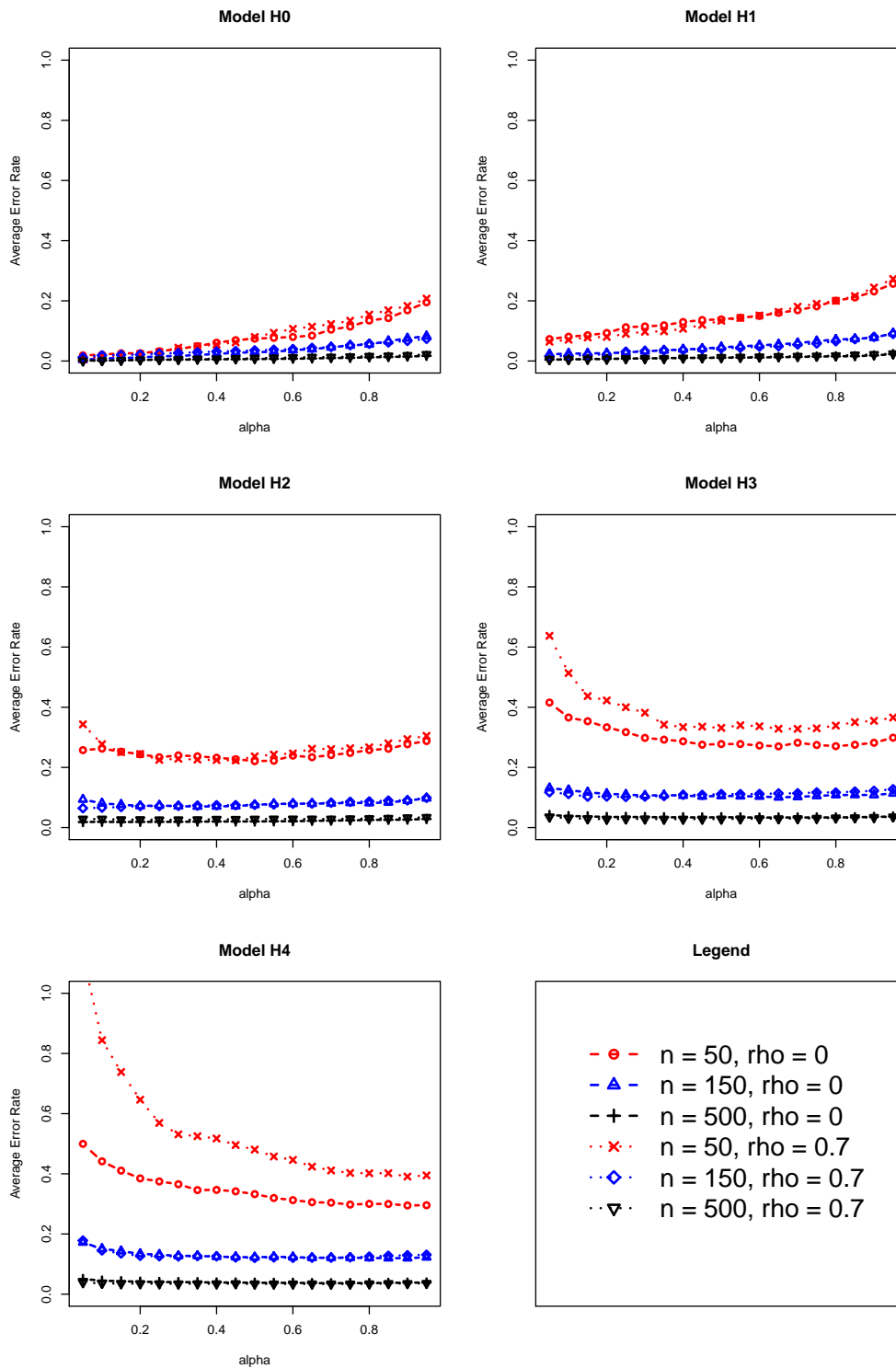


Figure 3.4: Average Model Error for different values of α

We notice for a large sample size, the choice of α is not that important, so we will concern ourselves more with the red ($n = 50$) and blue ($n = 150$) curves. Note the null H_0 and sparse H_1 models have error rates that are increasing in α . This is especially pronounced for the small sample size. The curve really steepens at $\alpha = .25$. On the other hand the relatively saturated H_3 and H_4 models have error rates that are decreasing in α . The decrease roughly plateaus around $\alpha = .30$. Model H_2 is relatively indifferent to the choice of α . Based on this, we recommend $\alpha = 0.20$ to avoid the gross errors which are evident for small α in Model H_4 , while still maintaining sparsity. We could conceivably choose α anywhere in the range 0.10 to 0.40. Note that $\alpha = 0.20$ corresponds approximately to the quantile of the maximum F -statistic for the RIC in the orthogonal predictor case.

3.9 Relation to the False Selection Rate

We mentioned that the False Selection Rate is similar to the PIC in the previous chapter. We now expand on this point. The idea of adding noise variables to compare variable selection methods is not new, see e.g. (Miller, 2002). However, the FSR and PIC differ from these methods because they use the noise variables directly to tune and select a model. Both methods augment the data set with predictors that have no relation to the response. If we do not have too many predictors and we use a relatively small FSR, we might expect the two methods to approximately coincide. For example, if we have 20 predictors and an FSR of .05, then we might expect to include at most $20 \times .05 = 1$ false

predictor.

We take a few issues with the FSR. First, the dichotomy of important $\beta \neq 0$ and unimportant $\beta = 0$ predictors is not completely fair. Often, even if a coefficient is nonzero, we are better off leaving it out of the model. The decrease in bias is not worth the increase in variance. This can happen when the coefficient is small, or in the presence of collinearity, its effect is already accounted for by other predictors.

Additionally, one downside to methods like the FSR and Cross-validation when it is used to select a model size is that we estimate the size from a different model than our final one. We mentioned this idea above in the Cross-validation section. For example, when using Forward Selection on one of the folds, there is no reason why the variables should enter in the same order, or even be the same variables, as they would on the full data set. Similarly, the FSR does not stop the first time it selects a noise variable. This means after we adjust for the noise variable, we face a random residual space. This may not be that pronounced for a small number of predictors p , but we believe in the large $p > n$ case, this could be severe. Our method naturally extends to the $p > n$ case. We also believe strongly that augmenting \mathbf{X}_π is the most natural way to augment for reasons mentioned above.

Chapter 4

Rotations vs. Permutations

We now extend the idea of augmenting the predictor space with permuted data to a more general family – rotated data. Consider the same setup as before with \mathbf{X} the actual data and \mathbf{X}_π the row-permuted data. Another way to express \mathbf{X}_π is

$$\mathbf{X}_\pi = \mathbf{P}\mathbf{X}$$

with \mathbf{P} a $n \times n$ permutation matrix with exactly one 1 in each row and column with the rest of the entries 0. Let p_{ij} be the entry in the i^{th} row and j^{th} column of \mathbf{P} . Then if $p_{ij} = 1$, the i^{th} row of \mathbf{X}_π equals the j^{th} row of \mathbf{X} .

One characteristic of permutation matrices is that $\mathbf{P}^T\mathbf{P} = \mathbf{I}$, or that its transpose is the inverse. This is precisely the algebraic reasoning why \mathbf{X}_π and \mathbf{X} share the same correlation structure¹.

$$\mathbf{X}_\pi^T\mathbf{X}_\pi = \mathbf{X}^T\mathbf{P}^T\mathbf{P}\mathbf{X} = \mathbf{X}^T\mathbf{X}$$

¹Assuming the data has been centered

The fact that $\mathbf{P}^T \mathbf{P} = \mathbf{I}$ is a property of a more general matrix family – orthogonal matrices. Let us denote an arbitrary orthogonal matrix by \mathbf{Q} . \mathbf{Q} is constructed by taking any set of n orthonormal vectors in \mathbb{R}^n and joining them as the columns of \mathbf{Q} . Consequently, \mathbf{QX} has the same correlation structure as \mathbf{X} . A permutation matrix is a special case of a rotation matrix.

We will adopt the notation of Langsrud (2005) and call a matrix \mathbf{Q} a rotation matrix.² We now consider when do permutations and rotations apply for valid inference? Since rotation matrices are more general than permutation matrices, we would expect less stringent assumptions on the data. We will see that the fundamental idea is the same – we generate datasets that have equal probability with respect to a null hypothesis. Consequently, any test statistics we construct, when ranked, are uniform and lead to exact p-values. In practice, we will never sample the entire distribution of permutations or rotations and thus our p-values will be exact up to simulation error.

Most every classical test statistic for multiple regression is a function of 3 quantities: $\mathbf{y}^T \mathbf{y}$, $\mathbf{X}^T \mathbf{X}$, and $\mathbf{X}^T \mathbf{y}$. This includes the standard t -statistics, as well as multivariate generalizations such as Hotelling's T^2 , Wilks' Lambda, Roy's largest root, and various trace statistics. As we showed above, $\mathbf{X}^T \mathbf{X}$, and similarly $\mathbf{y}^T \mathbf{y}$ is invariant to multiplication by \mathbf{Q} . The only quantity affected by the random rotation is $\mathbf{X}^T \mathbf{y}$. Suppose we rotate \mathbf{X} by multiplying by \mathbf{Q} . Then

$$(\mathbf{QX})^T \mathbf{y} = (\mathbf{X}^T \mathbf{Q}^T) \mathbf{y} = \mathbf{X}^T (\mathbf{Q}^T \mathbf{y})$$

²Technically, rotation matrices have determinant equal to 1. We also allow the determinant to be -1.

so we see that multiplying \mathbf{X} by \mathbf{Q} is equivalent to multiplying \mathbf{y} by \mathbf{Q}^T . As \mathbf{Q}^T is the inverse of \mathbf{Q} , \mathbf{Q}^T is also a rotation matrix. Consequently, whether we rotate \mathbf{X} or \mathbf{y} , or both is inconsequential.

4.1 When is Permutation Valid?

Suppose that we have an independent sample from a joint distribution $(\mathbf{y}_i, \mathbf{x}_i) \sim f_{Y,X}$ where $\mathbf{y}_i \in \mathbb{R}$ and $\mathbf{x}_i \in \mathbb{R}^p$, and $i = 1, 2, \dots, n$. Under the null hypothesis of independence between x and y we write

$$H_0 : f_{X,Y}(\mathbf{x}, \mathbf{y}) = f_X(\mathbf{x})f_Y(\mathbf{y})$$

where f_X and f_Y are the marginal distributions of \mathbf{X} and \mathbf{y} respectively.

Consequently, for any permutation $\pi \in \mathbf{\Pi}$,

$$f_{X,Y}(\mathbf{X}_\pi, \mathbf{Y}) = f_X(\mathbf{X}_\pi)f_Y(\mathbf{Y}) = f_X(\mathbf{X})f_Y(\mathbf{Y}) = f_{X,Y}(\mathbf{X}, \mathbf{Y})$$

because of the fact that $f_X(\mathbf{X}_\pi) = f_X(\mathbf{X})$ with independent data. Reordering the data does not affect the joint probability.

This means that from whatever distribution the data is generated, under the null hypothesis of independence, the permuted data has the same probability as the actual data. Consequently, any test statistic, such as the maximum F -statistic, constructed from the permuted data will have the same distribution as the test statistic constructed from the actual data. We are generalizing Pitman (1937) who derived this for the correlation between \mathbf{y} and a univariate \mathbf{x} .

4.2 When is Rotation Valid?

In multiple linear regression, probability enters through the error vector. We want to generate datasets by rotation that have the same probability as the original data. This means for any orthogonal matrix \mathbf{Q} , we have the relation

$$f_{\epsilon}(\epsilon) = f_{\epsilon}(\mathbf{Q}\epsilon)$$

That is, the error distribution is invariant with respect to rotations. This is equivalent to

$$f_{\epsilon}(\epsilon) = f_{\epsilon}(\|\epsilon\|)$$

or the error distribution is solely a function of its length. This is more commonly known as a *spherically symmetric distribution*.

Now under the null

$$H_0 : \mathbf{y} = \epsilon \quad \epsilon \sim f_{\epsilon}(\|\epsilon\|)$$

any test statistic $\hat{T}(\mathbf{X}, \mathbf{y})$ has the same distribution as $\hat{T}(\mathbf{QX}, \mathbf{y})$.

Consequently, we could alternatively augment the predictor space

$$\tilde{\mathbf{X}} = (\mathbf{X} \mid \mathbf{QX})$$

and proceed with the PIC as discussed in Chapter 3. The key difference lies in the assumptions we make. To permute, we require independent observations from an arbitrary error distribution. To rotate, we require a spherically symmetric error distribution.

We still use permutations in all that follows because they are cheaper to compute. For more on rotation tests see (Langsrud, 2005).

Chapter 5

Extending the PIC

5.1 Generalized Linear Models

Variable selection is most frequently discussed in the context of linear regression. In fact, predictive criteria like Mallows' C_p and RIC were developed specifically for linear regression. Information based criteria e.g., AIC and BIC, and Data-resampling methods e.g., cross-validation and the bootstrap, are more widely applicable. In this chapter, we show that the PIC is also widely applicable and show its use with Generalized Linear Models.

Generalized Linear Models (McCullagh and Nelder, 1989) extend linear models by relating a function of the mean of \mathbf{y} linearly to \mathbf{X} . This function is called the link function and we denote it by g . Members of the GLM family have the form

$$g(\mathbf{E}(\mathbf{y})) = \mathbf{X}\beta$$

Common examples include logistic regression, $g(\mathbf{z}) = \log(\mathbf{z}/(1 - \mathbf{z}))$ and Poisson regression $g(\mathbf{z}) = \log(\mathbf{z})$. Linear regression is simply the identity function $g(\mathbf{z}) = \mathbf{z}$.

When selecting the best variable under forward selection in linear regression, we choose the variable with the largest F-statistic. This is equivalent to selecting the variable that decreases error the most. In linear regression, the error is the residual sum of squares. In Generalized Linear Models, the error is the deviance. The deviance is defined as

$$D(\mathbf{y}) = -2 \left[\log(p(\mathbf{y}|\hat{\theta}_k)) - \log(p(\mathbf{y}|\hat{\theta}_n)) \right]$$

or -2 times the difference in log likelihoods between the current model fit with parameter vector denoted by $\hat{\theta}_k$ and the full model fit if we used a parameter for every observation with parameter vector $\hat{\theta}_n$. We use the subscript n to denote all n observations. The deviance subtracts the full model log-likelihood so that a deviance of 0 is meaningful.

5.2 PIC with GLM

The PIC can be extended to Generalized Linear Models easily with one slight modification. We begin by augmenting the predictor space \mathbf{X} with a row-permuted version of it \mathbf{X}_π just like before. We proceed with Forward Selection with the augmented space $\tilde{\mathbf{X}}$ and select the best variable at each step. As soon as the best variable comes from \mathbf{X}_π , we stop. We define the best variable to be the one that decreases the deviance the most. This is equivalent to the most significant variable from the log likelihood ratio test between two models – one with the variable and one without it. We mention this because GLM

has different test statistics for individual coefficients. For example, in logistic regression we can use the Wald test or the Score test for an individual coefficient. They both have asymptotic χ^2 distributions and while they give similar results, they are not the same. We might have ambiguity about which variable is best. Our choice of using the deviance directly coincides with linear regression, assuming σ^2 is known.

One key difference with GLM is we do not have the linear algebraic framework of linear regression with orthogonal residual spaces. Consequently, we do not adjust our predictors after each one enters the model. This presents a problem. Suppose again without loss of generality, the variables in \mathbf{X} are arranged as they would enter in Forward Selection (\mathbf{X}_1 enters first, etc.). One side effect of the linear regression adjustment scheme was that when we adjusted \mathbf{X}_π with respect to \mathbf{X}_{π_1} , the first permuted predictor $\mathbf{X}_{\pi_1:\pi_1}$ gets annihilated to the zero vector. This ensured that for a model with k variables, both \mathbf{X}^k and \mathbf{X}_π^k have $p - k$ remaining variables. The current proposal for PIC applied to GLM does not possess this trait. \mathbf{X}_π^k still has the full p variables for all k . Consequently, if the actual data \mathbf{X}^k has no remaining signal to explain \mathbf{y} , we would be more likely to select our next variable from \mathbf{X}_π^k which has more predictors than \mathbf{X}^k . Ideally, we want this probability to be equal between the two. Therefore, we propose a simple amendment. As soon as we select \mathbf{X}_k to enter the model, we drop \mathbf{X}_{π_k} from further consideration. This ensures that \mathbf{X}^k and \mathbf{X}_π^k have the same number of predictors at each stage of Forward Selection. Moreover, their correlation structure directly coincide for each k . With this slight modification, we proceed as before.

The remaining details of how to select the final model remain the same as linear regression. We refer the reader to Section 3.7

Chapter 6

Simulation Results

We now apply the Permuted Inclusion Criterion to various simulations and compare it with well-known selection techniques.

6.1 Simulation Setup

We begin studying the performance of the PIC through a wide range of Monte Carlo simulations. We mimic the setup of Wu et al. (2007), which was initially used by Tibshirani and Knight (1999a). Every model has 21 predictors generated from a multivariate normal distribution with mean 0 and autoregressive covariance structure

$$\text{cov}(x_i, x_j) = \rho^{|i-j|}$$

We consider $\rho = 0$ and $\rho = 0.7$. We center all predictor variables as well as the response so that we do not have to worry about an intercept. The nonzero coefficients are clustered

around \mathbf{X}_7 and \mathbf{X}_{14} with values given by

$$\beta_{7+j} = (h - j)^2 \quad \text{and} \quad \beta_{14+j} = (h - j)^2 \quad \text{for} \quad |j| < h$$

We consider $h = 0, 1, 2, 3,$ and 4 and denote these models $H_0, H_1, H_2, H_3,$ and H_4 . These correspond to models with $0, 2, 6, 10,$ and 14 nonzero coefficients, respectively.

We illustrate these coefficients for $H_1, H_2, H_3,$ and H_4 below. H_0 is the null model.

For each model we scale the coefficients so that the theoretical R^2 is 0.75 where

$$R^2 = \frac{(\mathbf{X}\beta)^T(\mathbf{X}\beta)}{(\mathbf{X}\beta)^T(\mathbf{X}\beta) + n\sigma^2}$$

We also consider a varying number of observations with $n = 50, 150,$ and 500 . In total, we consider 2 different correlation structures on 5 different models for 3 different sample sizes yielding 30 different models. We simulate each model 50 times and take the average error rate and average model size.

Next, we discuss our selection schemes. For each model we select by 9 different methods: AIC ($\lambda = 2$), BIC ($\lambda = \log(n)$), RIC ($\lambda = 2 \log(p)$), modified RIC ($\lambda = 2 \log(p/k)$), PIC ($\alpha = 0.20$), False Selection Rate ($\gamma_0 = .05$), 5-fold Cross-validation, the Lasso, and the Random Oracle.

For AIC, BIC, RIC, and modified RIC, we stop the first time the maximum F -statistic is below λ . For 5-fold Cross-validation, we select the model with the smallest cross-validated sum of squares. To select the ‘‘best’’ model, we adopt the Random Oracle idea (Benjamini and Gavrilov, 2009). The random oracle assumes that we know the true β

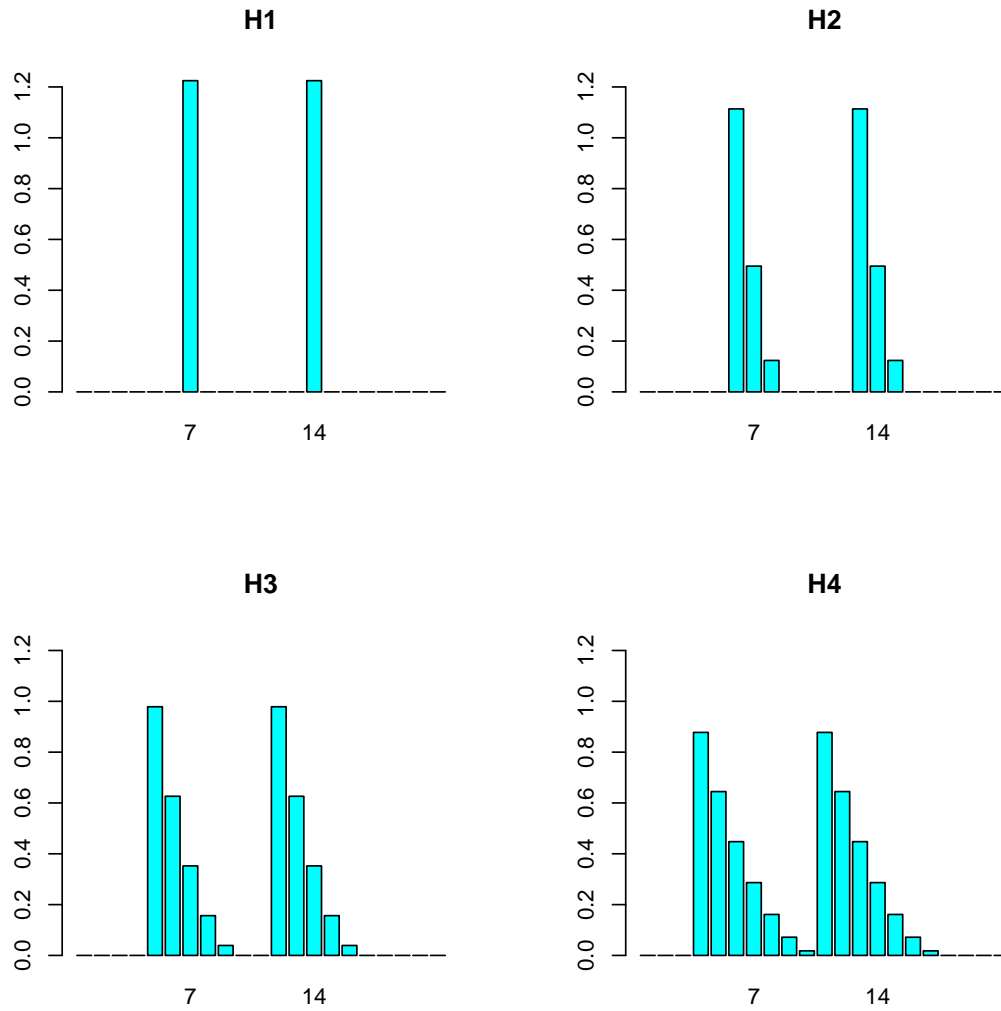


Figure 6.1: Regression coefficients for $\beta_1, \beta_2 \dots \beta_{21}$ for the 4 different models ($\rho = 0$)

vector. Consequently, we directly calculate the model error as

$$ME(\hat{\mu}_k) = \|\hat{\mu}_k - \mu\|^2 = \|\mathbf{X}\hat{\beta}_k - \mathbf{X}\beta\|^2$$

since we know β and consequently μ exactly. We just restrict ourselves to the $p + 1$ models generated by Forward Selection, and select the best one. Note that every selection

scheme, other than the Lasso, that we consider also selects a model from Forward Selection. Consequently, the Random Oracle will never have a larger error rate. The Lasso, however, could outperform the Random Oracle.

Because the plots of 9 curves gets cluttered, we broke the average error and average model size plots into 3 different sets of plots. First, we look at how the PIC compares to the False Selection Rate. The error curves, adopted from Wu et al., are defined as the average error for the Random Oracle divided by the average error for the given procedure. Consequently, the closer the curve is to 1, the better. For the error curves, we disregarded model $H0$ because often the oracle error rate was 0. The PIC and FSR give strikingly similar results. For a large sample size they virtually coincide. In terms of model size, again they are strikingly similar. The main difference is in model type $H4$, the PIC prefers more parsimonious models. These are models with many small coefficients.

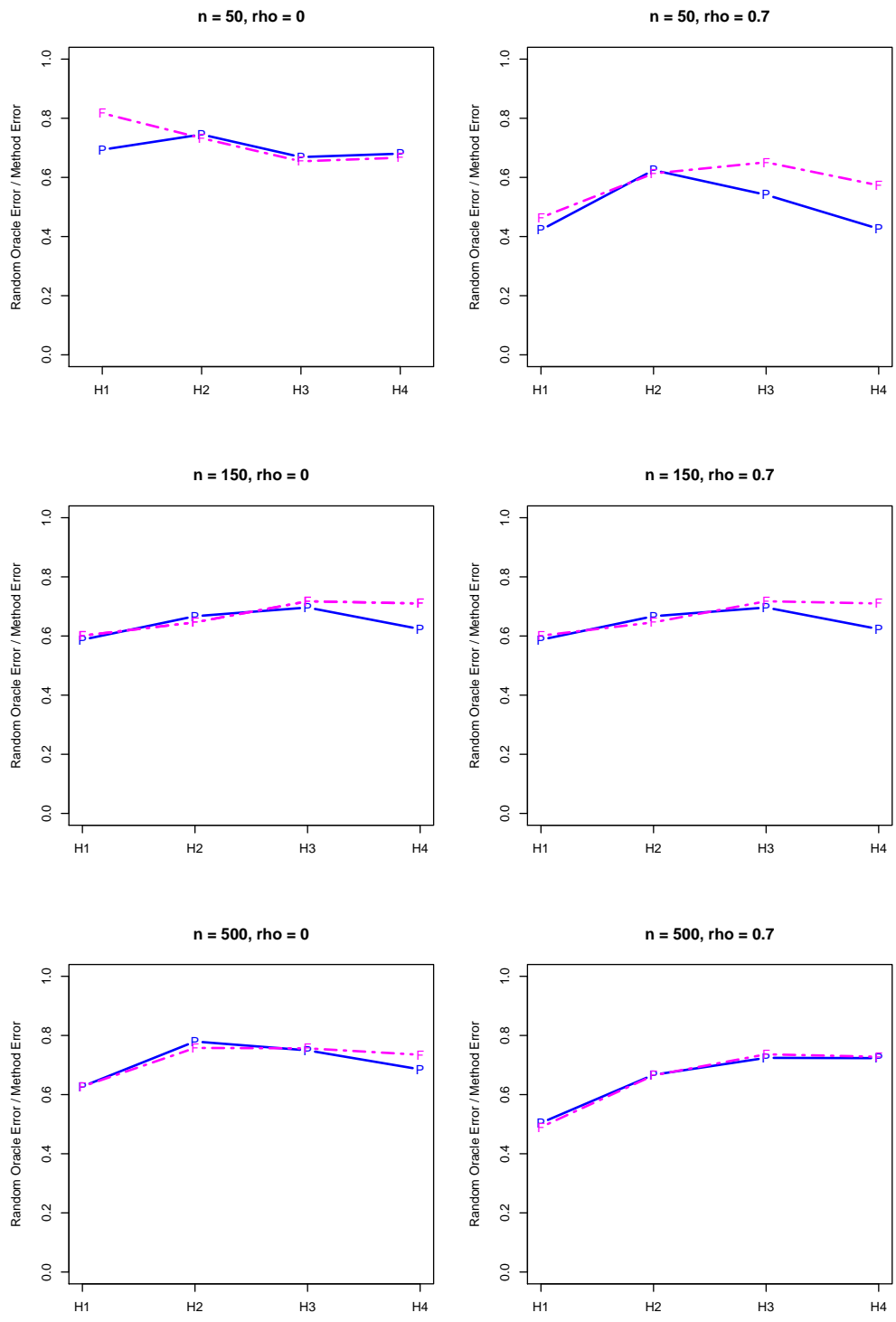


Figure 6.2: Average Error Rate for the False Selection Rate and the Permuted Inclusion

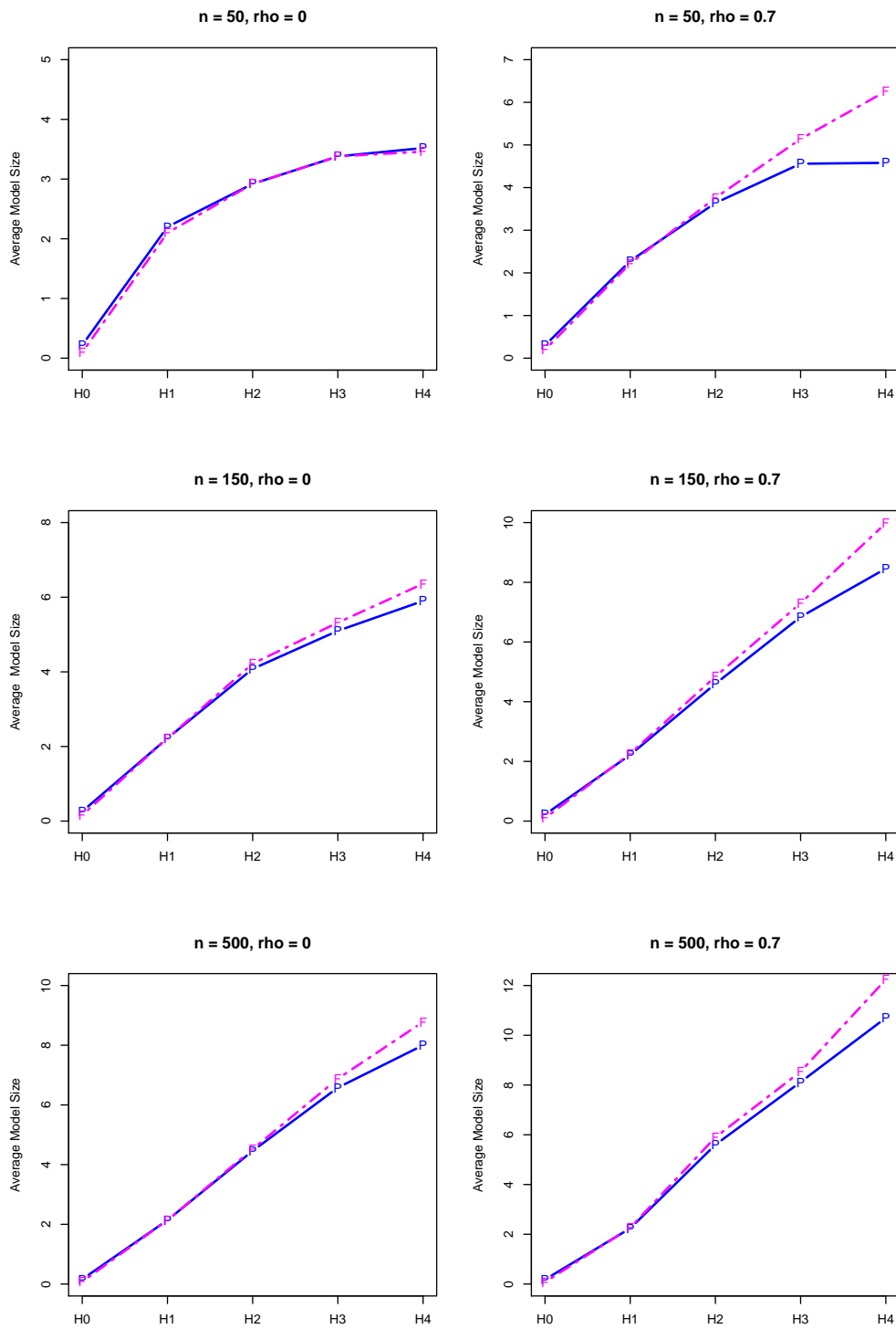


Figure 6.3: Average Model Size for the False Selection Rate and the Permuted Inclusion

Criterion

Next we analyze how the PIC compares to AIC, BIC, and RIC. We lump these selections schemes together because they use a λ penalty. The PIC and RIC behave quite similarly as we might expect. Both penalties grow with p . As we move down the plots, the sample size is increasing and consequently, the BIC is enforcing a harsher penalty. For $n = 500$, the PIC, RIC, and BIC practically coincide. As we might expect, the AIC performs quite differently from the other selection schemes. Not surprisingly, the AIC performs worst when we have a sparse model ($H1$) and does the best when our model is saturated ($H4$). In terms of model size, we see what we would expect. AIC always selects the most variables while BIC converges to where the PIC and RIC lie as n increases. The PIC and RIC have the same average model size in almost every point of every curve. The notable exception is model $H4$ when $n = 50$ and $\rho = 0.7$.

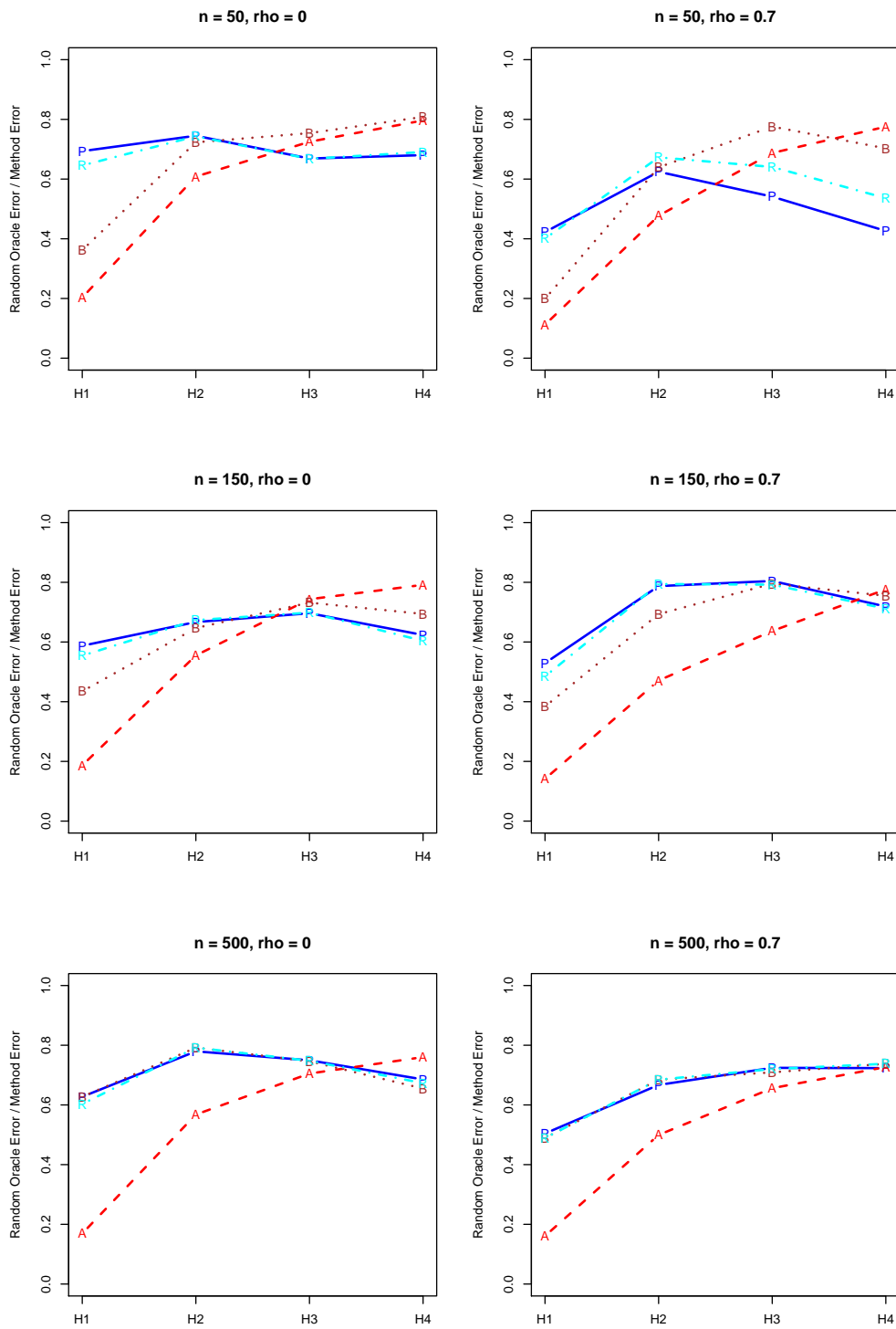


Figure 6.4: Average Error Rate for the PIC, AIC, BIC, and RIC

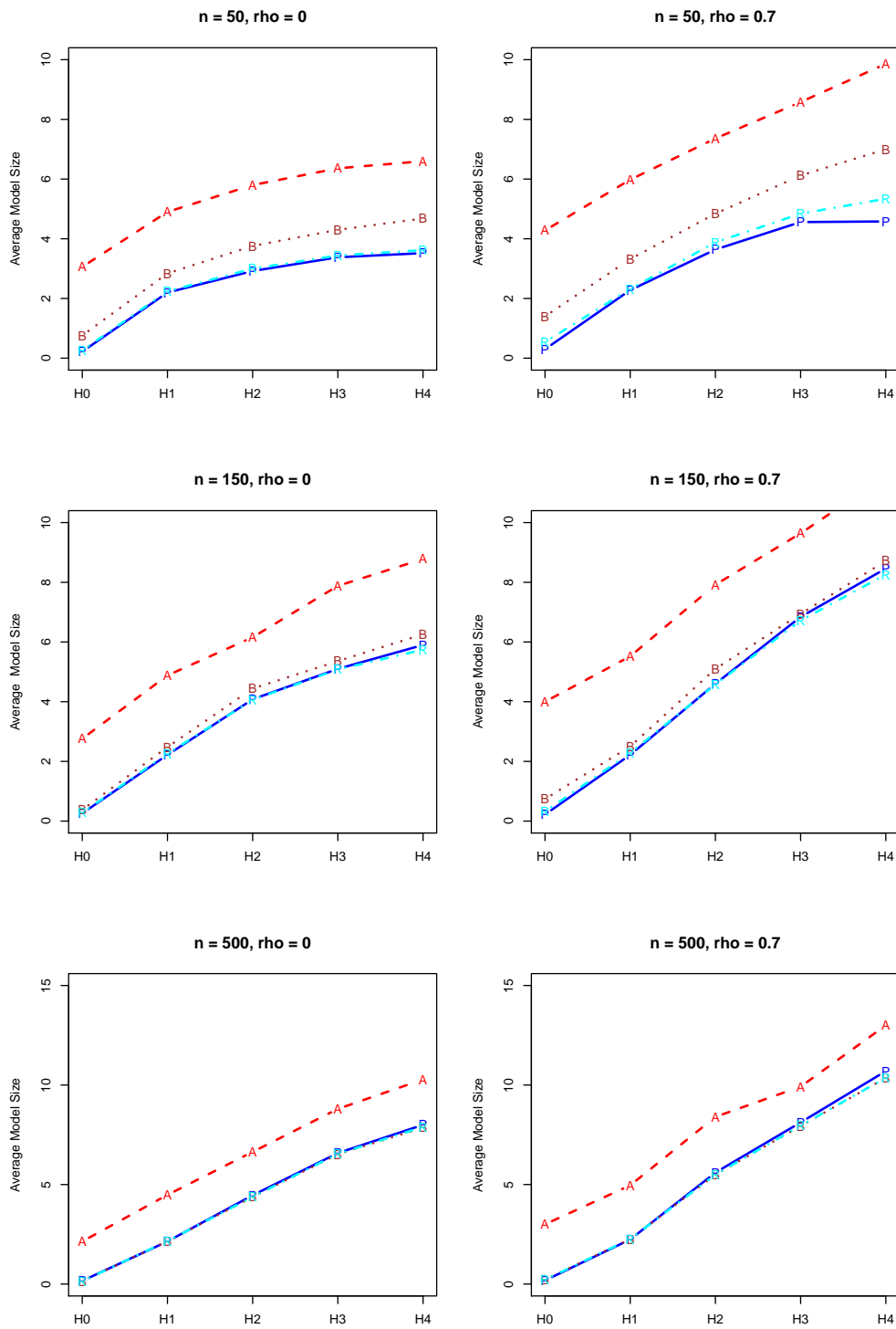


Figure 6.5: Average Model Size for the PIC, AIC, BIC, and RIC

Lastly, we analyze how the PIC compares to 5-fold Cross-Validation and the Lasso. We lump these two together because they are most unlike the other schemes. For model error, all 6 of these plots have a similar pattern. The PIC tends to dominate for models $H1$ and $H2$, but loses out for the more dense model $H4$. This is especially true in the correlated small sample data. In terms of model size, we have a uniform ranking over all of the plots with the PIC preferring the most parsimonious model and the Lasso selecting the most predictors. Cross-validation lies somewhere in between with similar model sizes to the PIC for $H1$ and $H2$ and less so for $H4$

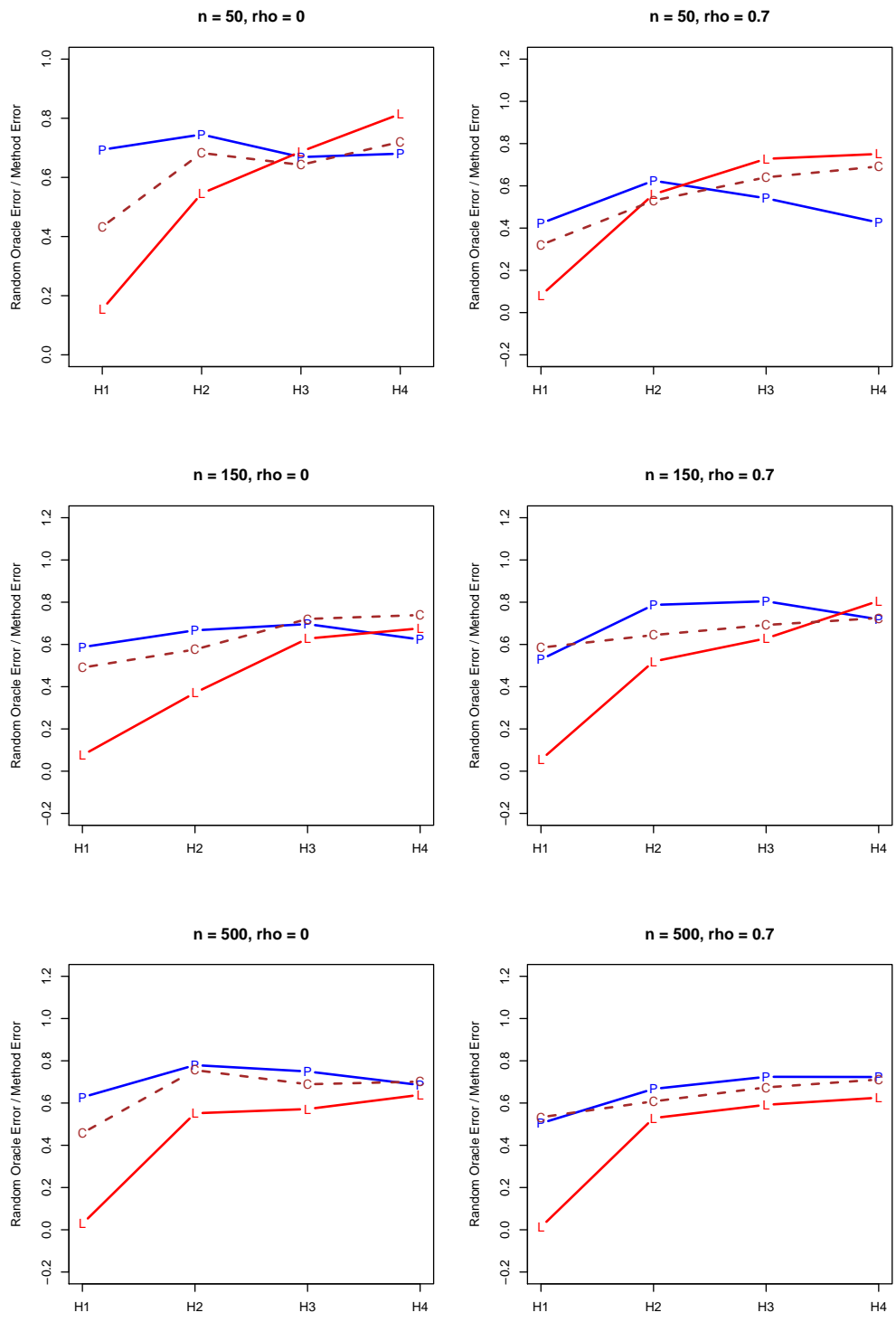


Figure 6.6: Average Error Rate for the PIC, 5-fold Cross-Validation and the Lasso

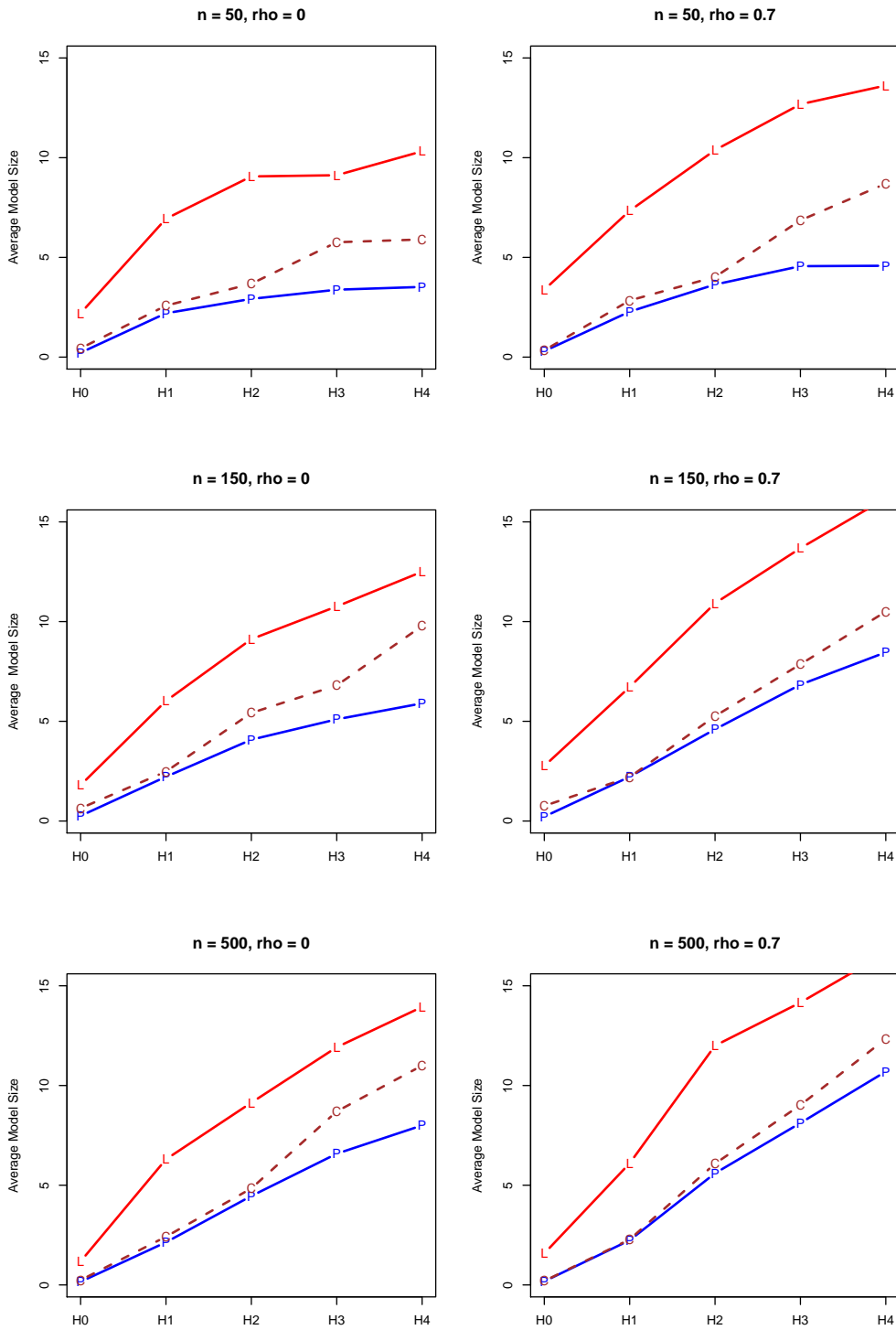


Figure 6.7: Average Model Size for the PIC, 5-fold Cross-Validation and the Lasso

6.2 Simulations: $p > n$

In this section, we investigate the performance of the PIC in a case where the number of variables is much larger than the number of observations. We use the same setup as the Candes & Tao paper with 72 observations on 256 variables where \mathbf{y} and $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{256}$ are generated as independent standard normal variables. The \mathbf{X} variables are scaled to have unit norm. We have 8 nonzero variables and we set a noise standard deviation of $1/9$.

Method	Model Error (SE)	Average Model Size (SE)
PIC ($\alpha = .20$)	.0021 (.00021)	8.48 (.119)
RIC	.0021 (.00021)	8.44 (.111)
mRIC	.0074 (.0040)	16.1 (.651)
RandOrac	.0014 (.00008)	8 (0)
DS	.1639 (.0043)	7 (.1639)
DS-Gauss	.0308 (.0026)	7 (.1639)

Table 6.1: Average Model Error and Model Size for a wide dataset $p = 256, n = 72$. We note the extremely similar performance between the PIC and RIC.

The PIC and RIC performed almost exactly the same only differing in 2 of the 50 simulations. Their error rate was not far from the Random Oracle either. We see that the Dantzig Selector performs poorly relative to the other methods. This was noted by Candes and Tao in their original paper where the selected variables exhibited a soft thresh-

olding behavior. Consequently, they proposed a two-stage procedure where the Dantzig Selector selects the variables to include and then we perform ordinary least squares on that subset. This is the DS-Gauss line above. This cut the average error rate by more than a factor of 5. We remark that we also considered AIC in these simulations but it nearly always selected all 72 predictor variables. We also note that in the $p > n$ case, the modified RIC does not perform that well. Too often, it was including irrelevant predictor variables.

Chapter 7

Permuted Selection and Trees

In this chapter, we apply the Permuted Inclusion Criterion to a very different family of models – Classification and Regression Trees (CART) (Breiman et al., 1984).

7.1 CART

CART is a nonparametric technique that recursively partitions the predictor space into rectangles by using binary splits and predicts a constant within each rectangle. We define the node being split as the parent node, and the two resulting nodes, the child nodes. We begin CART with all of the data together at the top of the tree, known as the root node. CART then searches over all possible split points of all possible variables and selects the best one. Intuitively, the best split is the one that makes the data within the same child node as similar as possible, while making the data in different child nodes as different as possible. Each resulting child node now becomes a parent node and in turn gets split

into two child nodes. This is the recursive nature of CART. For example, suppose CART selects $X_1 < 3$ as the best split point. Then all observations satisfying this inequality are sent down the tree to the left child node while the other observations not satisfying the inequality are sent to the right child node. Then, we repeat within each node. In the left child node, we select $X_2 < 4$ as the best split, while in the right child node, we select $X_2 < 2$ as the best split. We now have 4 terminal nodes which correspond to rectangles. Perhaps the greatest benefit of CART is its highly interpretable visual representation as a tree. Here is an example of the model that we just described, both as rectangles and as a tree.

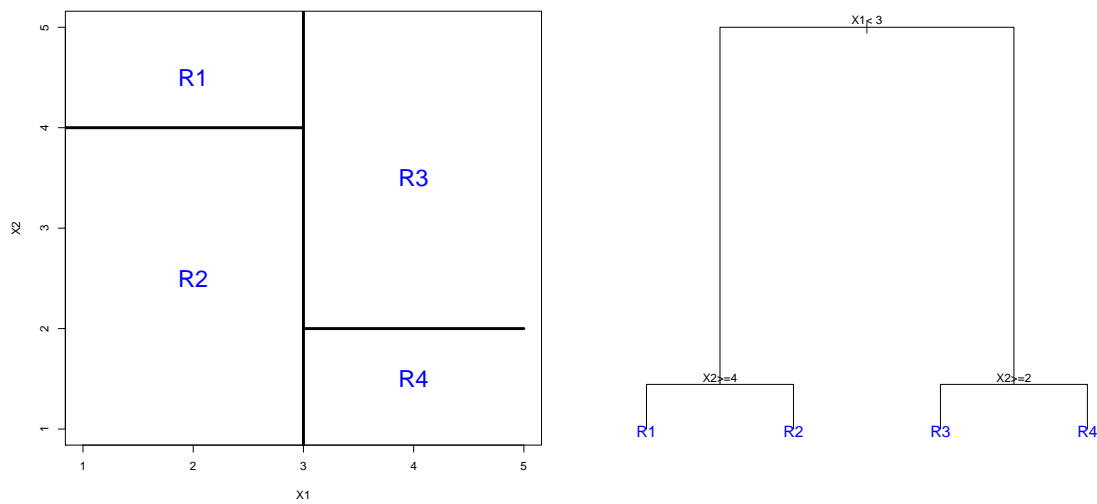


Figure 7.1: CART: Displayed as rectangles on the X_1 - X_2 plane on the left and as a Tree on the right

The tree is much more interpretable. We ask simple yes/no questions and we follow a path down the tree based on the answer. Its representation is popular among the medical

community as Hastie et al. (2001) note “because it mimics the way that a doctor thinks.”

CART is suited for both continuous responses and categorical responses. When we deal with continuous responses, we typically use squared error as our error measure. Consequently, our prediction for a given node is the mean of the data falling within it. For categorical responses, we typically choose either misclassification rate or a smoother differentiable criteria, such as the Gini index or relative entropy. Our prediction for a given node is the modal category in that node.

CART, just like Forward Selection, proceeds in a greedy fashion. At each step we choose the best split point, subject to some predefined constraints. One of the biggest questions with CART, like Forward Selection, is when do we stop? Unlike the linear model in regression where coefficients have distributional theory justified from the Central Limit Theorem, CART has no inferential basis. Suppose we have a continuous response variable. By using a measure like residual sum of squares, every split results in a decrease. We need a sensible way to know when to stop growing the tree.

7.1.1 Traditional CART Stopping Criteria: Pruning

The most common stopping criteria for CART is to purposely overgrow the tree, and then sequentially prune nodes. We use *weakest link pruning* by successively collapsing the non-terminal node that results in the smallest increase in error. Each time we collapse, we create a new subtree. We continue until all that remains is the root node. We then use cross-validation to select which tree among all subtrees predicts best. We propose an

alternative method to select a tree.

7.1.2 Permuted CART Stopping Criterion

The idea of augmenting the predictor space to mitigate selection bias is not unique to linear regression. In fact, this idea can be applied to any algorithm where decisions involve selecting a “best” variable. As CART searches over all split points of all possible variables, it naturally falls under this framework. As before, we augment the predictor space \mathbf{X} with a row-permuted version of it called \mathbf{X}_π .

Our algorithm is simple. We build a CART tree with the augmented predictor space. At each node we search over all split points, both from \mathbf{X} and \mathbf{X}_π . If the best split is real, we select that split and continue the CART algorithm. This is the same split point that would have been selected had \mathbf{X}_π not been part of our data. However, if the best split is from \mathbf{X}_π , we do not split that node further. That node becomes a terminal node. Once this occurs for every node, we have our CART tree. As this clearly depends on the realized permutation, we simulate this many times to create an entire distribution over nodes and measure how frequently it is split.

This provides an added benefit not typically available to CART – a quantitative measure of how certain we are about a split. We know how many of the trees reach a given node and how many of those trees further split that node by selecting a real predictor. For example, suppose we simulate 100 trees and 90 of them reach a given node. Additionally, suppose that 82 of those 90 trees select a real predictor. Then we have a sample

proportion of 82/90 trees selecting from \mathbf{X} instead of \mathbf{X}_π – strong but not overwhelming evidence that the split is real. We have this measure for every node. Traditional CART just splits nodes without any measure of how strong the split is. In linear regression with Forward Selection, we have p -values as a measure of strength for each variable, even if the p -value has no interpretation as a probability because of selection bias.

Since we always stop before stepping into permuted space, the trees are nested in the sense that every node either selects the exact same split point as other trees, is a terminal node because it selected a permuted predictor, or was never reached because one of its ancestors selected a permuted predictor.

7.1.3 How to Adjust and Select a Model

Recall that with linear regression, we had a choice of how to adjust the variables after each step. We have to make a similar choice with CART, and unlike linear regression, the most natural choice is to re-permute after each split. The clearest way to motivate why is with an example. Suppose that we have a categorical predictor like gender which has two categories: Male and Female, and that we select it as the first split. As a result, one child node has only Male for its gender, while the other child node has only Female. Consequently, neither node can be further split on gender. Now consider the permuted space \mathbf{X}_π and the permuted version of gender. In both child nodes the permuted gender has both Male and Female observations and so it could be selected as a split point further down the tree. \mathbf{X}_π has more variables to select splits from than does \mathbf{X} . This is undesir-

able. More generally, consider a continuous predictor. If we re-permute at each node, then the potential split points for \mathbf{X} directly coincide with the potential split points for \mathbf{X}_π for every node. This is the most natural setup and the one we will adopt. It also supports one of the main motivations for the PIC. Under a null assumption that \mathbf{X} has no relation to \mathbf{y} , selecting from \mathbf{X} versus \mathbf{X}_π should be equally probable.

Lastly, we select our CART model the same way we select our regression models. We specify a proportion α as our cutoff. Our selected tree consists of precisely of those nodes that appear in at least $1 - \alpha$ of the trees and excludes all other nodes. This is best shown with an example.

7.1.4 Example

We illustrate the PIC applied to CART with two examples. First, we consider the classic Boston housing data (Harrison and Rubinfeld, 1978) taken from 506 census tracts around the Boston area for the 1970 Census. The goal of this data set is to predict the median housing price based on 13 covariates. These variables are:

Variable	Description
CRIM	Per capita crime rate
ZN	Proportion of residential land zoned above 25,000 sq ft
INDUS	Proportion of non-retail business acres
CHAS	Dummy variable for on Charles River
NOX	Nitric Oxide concentration (parts per 10 million)
RM	Average number of rooms
AGE	Proportion of units built prior to 1940
DIS	Weighted distance to 5 Boston employment centers
RAD	Index of accessibility to radial highways
TAX	Property tax per \$10,000
PTRATIO	Pupil to teacher ratio
B	$1000(B - 0.63)^2$ where B is Black proportion
LSTAT	Percentage of lower status in population
MEDV	Median value of homes

Table 7.1: Boston Housing Data taken from Harrison and Rubinfeld

We built a CART tree with $\alpha = 0.10$. For each split point, we also include the fraction of augmented data sets that selected the real split point, instead of one of the permuted predictors. The terminal nodes give the predictions which are the mean of the response for observations falling in that node.

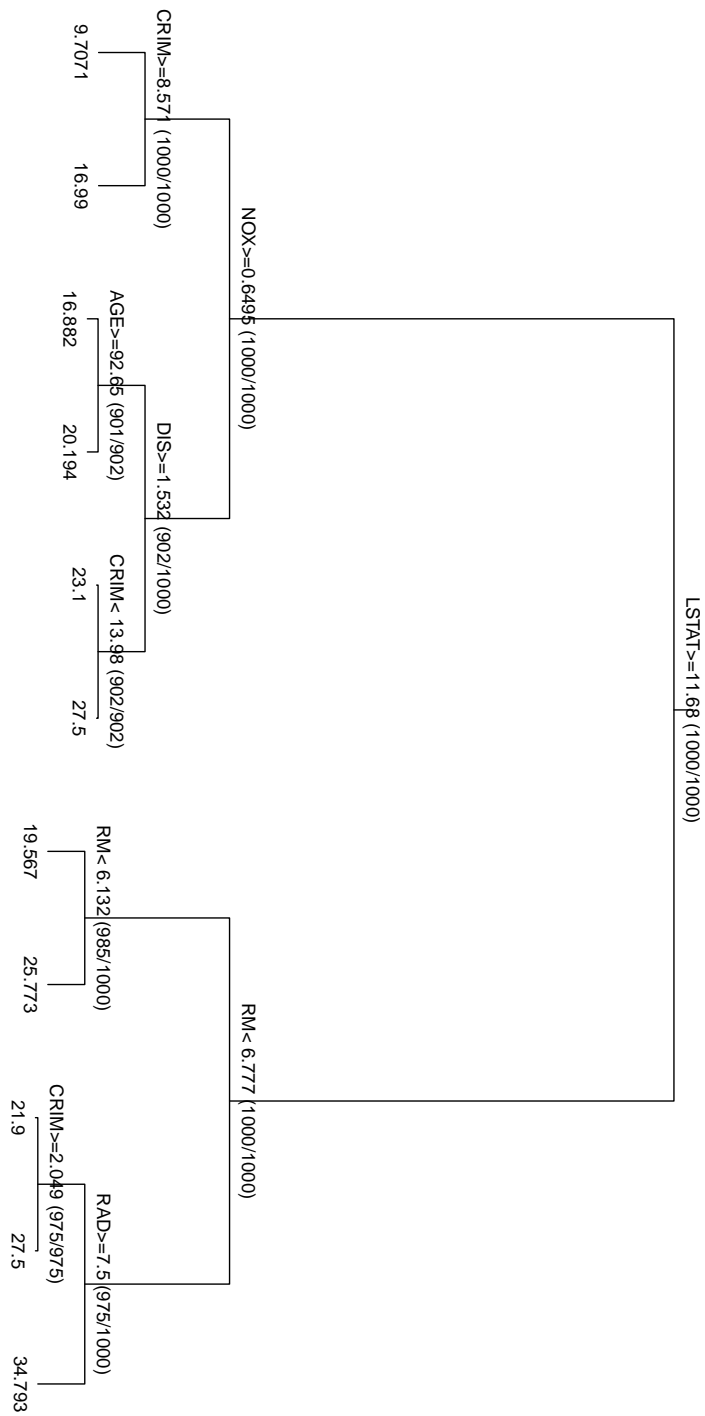


Figure 7.2: CART with PIC for the Boston Housing Data. $\alpha = 0.10$

We see that for the first three split points all 1,000 of our trees selected the actual splits – overwhelming evidence that these splits are real. If we look at the split for DIS on the left hand side, we are relatively less certain about this split point with 902 of the 1000 trees selecting this split. We also see a common manifestation of greedy algorithms. Although we are less sure about the DIS split, once we select it, its two child nodes are selected 901/902 and 902/902 times – almost always. We would sometimes like to look two or three steps ahead with greedy algorithms. We see here the best one step ahead move was not overwhelming but once we selected it, its next moves were.

We next illustrate the PIC on a complete noise example. We have 100 observations on 10 variables where the response y and each predictor variable X_1, X_2, \dots, X_{10} is generated as a standard normal random variable. We also set a very large $\alpha = 0.995$ so that we can see the fully grown tree and the sample selection proportions for each node. This means we show all nodes that appeared in at least 5 of the 1,000 trees.

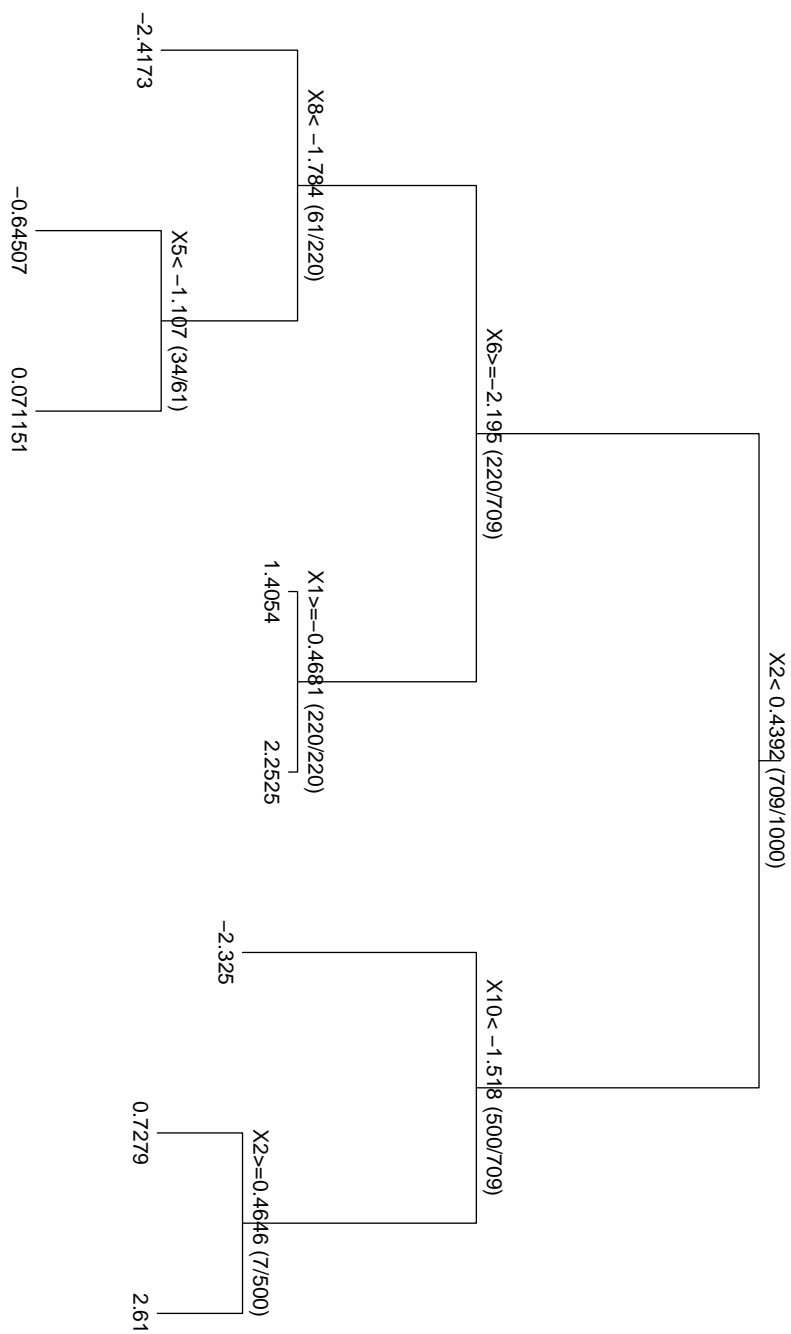


Figure 7.3: CART with PIC for complete noise data. $\alpha = 0.995$

We see that none of the splits have selection proportions near 1 except for the split on X_1 in the middle of the plot. Again, we see an illustration of how one-step ahead greedy algorithms can behave. Although the split on X_6 appears in only 220 out of 709 trees, once we select that split, every single one of the 220 trees selected X_1 . We also note that none of its child nodes were selected for a split meaning less than 5 of the trees selected a real split. We also see another dramatic example on the right side. Of the 500 trees that reach the split on X_2 , only 7 trees selected that split.

7.1.5 Choice of α

The choice of α for CART has no relation to the choice of α for Forward Selection. A key difference with Forward Selection is that we stop as soon as we would step into permuted space. With CART, we do not stop until every terminal node would next step into permuted space. Consequently, we might imagine choosing a smaller α much like we did with the Boston housing data.

However, we instead advocate the PIC with CART primarily as a diagnostic tool to quantify the certainty of splits. We believe this is its greatest benefit because CART is nonparametric and does not have a measure like this. We could even use weakest link pruning to select our final tree and use the PIC to measure each split. Alternatively, we could use cross-validation to select α .

7.2 Tree Extensions

While CART gives highly interpretable graphical representations, it's no longer state-of-the-art in terms of performance. However, many modern algorithms that are state-of-the-art use CART as a building block in ensemble methods that build a collection of trees and average them. We highlight three of those algorithms: Bagging (Breiman, 1996), Random Forests (Breiman, 2001), and Boosting (Freund and Schapire, 1996).

Bagging starts by taking bootstrap samples of the data, and for each bootstrap sample we build a CART tree. Typically we let each tree grow fully. Note that since each bootstrap sample has around 63% of the observations, we will not predict perfectly. We build B trees, e.g. $B = 500$, and to predict, we take the predictions from each tree and average them. The key intuition for why bagging works is that trees are highly variable. Any split that occurs at the top of the tree propagates down to all future splits. Suppose at the top of the tree two splits are equally good. The trees that result should we choose one split over the other can be vastly different. Consequently, our predictions can vary greatly. By injecting randomness via the bootstrap we create multiple trees so that we can sample both splits. By taking an average, we decrease this large variance at the expense of an increase in bias. Typically, this decrease in variance more than offsets the increase in bias, and performance improves.

Random Forests can be viewed as an extension to Bagging where we try to decrease the correlation between trees. Random Forests begins just like the Bagging by taking bootstrap samples and building CART trees fully. The key difference takes place at the

split points of the trees. In Random Forests, we take a random sample of the predictor variables at each split point and select the best among them. This allows us to explore tree space more fully. The trees are less correlated than they were in bagging because of this extra randomness and consequently, have smaller variance. Random Forests have two sources of randomness: the bootstrap, and the random predictors.

Boosting attacks the problem from a different direction. Boosting has many variants. We will focus on Adaboost. Suppose we have a categorical response. Adaboost starts by growing a simple tree, often a stump which consists of a single split, and then predicts the response y . Points that are classified correctly are down-weighted while misclassified points are up-weighted. We then repeat the process and fit another simple tree to this weighted data and again down-weight or up-weight. After we have built many trees, e.g. 500, our final prediction is then based on a weighted average of the trees.

We only briefly touched on these methods to illustrate one key point: they all use trees as a building block. However, they could not be more different as to how they build them. Both Bagging and Random Forest grow trees fully. Even though they are based on a bootstrap sample, the trees have on the order of $2/3n$ terminal nodes. On the other hand, boosting creates trees with a small number of terminal nodes, frequently as small as two. This is even more striking when we consider Breiman's conjecture in his original Random Forest paper that Adaboost is a Random Forest. The fact that these two ensemble methods, properly tuned, perform so well, yet have tree sizes on opposite ends of the spectrum, makes me wonder whether we can more intelligently choose the

tree size. If we let the data adaptively choose the size of the tree, like the PIC applied to CART, perhaps we can obtain even better performance.

Chapter 8

Conclusions

We have developed a new framework for variable selection that possesses close ties to permutation testing theory and asymptotically coincides with the Risk Inflation Criterion.

We have seen its performance is on par with many other well-known variable selection techniques and additionally performs well in the $p > n$ case. The PIC can be viewed as a data-adaptive penalty where the penalty is increasing in p and decreasing in k coinciding with many state of the art model selection schemes.

Additionally, we have extended our method to generalized linear models and to a framework not typically viewed as variable selection – building CART trees. We have added a new dimension to CART by quantifying how certain we are about splits.

The fundamental idea behind the PIC is to generate a reference data set that has no relation to the response. This reference data set should have all of the same options available to it as the real data set. In the linear regression context, this meant the same number

of variables, $p - k$ at the k th step of Forward Selection that has the same covariance structure as the real data. For CART, this meant the same possible split points at each node. Thus, if the real data has no signal remaining, our decision between the real and fake data should be approximately equal. Additionally, the algorithm must stop whenever the next best variable is a permuted variable. By not stepping into permuted space, the algorithm selects one of the models that would be generated if run on the real data alone. Viewing the PIC as a data-adaptive penalty for greedy-algorithms that involve selecting a “best” variables leaves much room for future work. For example, we could apply the PIC to the Least Angle Regression family as an alternative to the C_p type penalty. We have also thought how we might generalize this to backwards elimination. For example, we might consider dropping a variable if we were to replace it with a permuted version of itself and the model fit actually improves. This scheme does not create nested model, however, because we may have a choice about which variables to drop, influencing the next choice. We can not just stop the first time we drop a real variable. If we could develop a backwards version of this scheme, then we could naturally extend it to Stepwise regression where we add and delete variables. This also falls under the framework of the Lasso. We could extend the PIC with CART trees by using them in Random Forests or Boosting. Because α controls the size of the trees, we might choose a small α with boosting and a large α with Random Forests.

Appendix A

Simulation Results

ρ	Method	H0	H1	H2	H3	H4
0	AIC	0.2003	0.2206	0.2821	0.3131	0.3175
	BIC	0.0817	0.1238	0.2373	0.3010	0.3128
	RIC	0.0366	0.0694	0.2309	0.3398	0.3661
	mRIC	0.0471	0.1419	0.2628	0.3186	0.3229
	PIC	0.0312	0.0646	0.2303	0.3394	0.3717
	FSR	0.0172	0.0549	0.2342	0.3468	0.3793
	CV	0.0383	0.1034	0.2514	0.3533	0.3513
	Lasso	0.0256	0.2908	0.3145	0.3302	0.3105
RandOrac	0.0000	0.0448	0.1716	0.2270	0.2529	
0.7	AIC	0.2719	0.2828	0.3135	0.3517	0.3855
	BIC	0.1391	0.1593	0.2339	0.3116	0.4253
	RIC	0.0695	0.0791	0.2223	0.3774	0.5577
	mRIC	0.0814	0.1931	0.3162	0.3681	0.4383
	PIC	0.0426	0.0750	0.2395	0.4467	0.7014

ρ	Method	H0	H1	H2	H3	H4
0	AIC	0.0684	0.0844	0.0902	0.1136	0.1239
	BIC	0.0187	0.0362	0.0774	0.1153	0.1414
	RIC	0.0150	0.0284	0.0744	0.1209	0.1624
	mRIC	0.0175	0.0559	0.0819	0.1165	0.1292
	PIC	0.0142	0.0268	0.0751	0.1212	0.1574
	FSR	0.0097	0.0262	0.0775	0.1177	0.1382
	CV	0.0220	0.0321	0.0870	0.1171	0.1327
	Lasso	0.0081	0.2088	0.1350	0.1345	0.1451
	RandOrac	0.0000	0.0157	0.0501	0.0844	0.0981
	0.7	AIC	0.0922	0.0968	0.1125	0.1063
BIC		0.0294	0.0357	0.0764	0.0851	0.1367
RIC		0.0152	0.0283	0.0667	0.0855	0.1450
mRIC		0.0254	0.0654	0.1151	0.1209	0.1389
PIC		0.0111	0.0259	0.0673	0.0842	0.1436
FSR		0.0056	0.0274	0.0731	0.0910	0.1353
CV		0.0172	0.0234	0.0822	0.0978	0.1423
Lasso		0.0089	0.2539	0.1020	0.1077	0.1280
RandOrac		0.0000	0.0137	0.0530	0.0677	0.1032

Table A.2: Average Error Rate (n = 150)

ρ	Method	H0	H1	H2	H3	H4
0	AIC	0.0148	0.0222	0.0250	0.0349	0.0361
	BIC	0.0022	0.0060	0.0179	0.0329	0.0419
	RIC	0.0022	0.0062	0.0179	0.0329	0.0408
	mRIC	0.0022	0.0143	0.0238	0.0373	0.0383
	PIC	0.0025	0.0060	0.0182	0.0328	0.0401
	FSR	0.0014	0.0060	0.0187	0.0325	0.0374
	CV	0.0027	0.0082	0.0188	0.0357	0.0392
	Lasso	0.0013	0.1320	0.0257	0.0430	0.0430
	RandOracle	0.0000	0.0038	0.0142	0.0246	0.0275
0.7	AIC	0.0210	0.0243	0.0309	0.0326	0.0406
	BIC	0.0037	0.0080	0.0225	0.0302	0.0400
	RIC	0.0037	0.0080	0.0225	0.0297	0.0400
	mRIC	0.0041	0.0155	0.0335	0.0383	0.0436
	PIC	0.0032	0.0078	0.0231	0.0296	0.0408
	FSR	0.0017	0.0080	0.0232	0.0291	0.0405
	CV	0.0032	0.0073	0.0254	0.0318	0.0414
	Lasso	0.0016	0.3014	0.0292	0.0362	0.0471
	RandOracle	0.0000	0.0039	0.0154	0.0214	0.0295

Table A.3: Average Error Rate (n = 500)

ρ	Method	H0	H1	H2	H3	H4
0	AIC	3.06	4.90	5.80	6.36	6.60
	BIC	0.76	2.84	3.76	4.30	4.68
	RIC	0.26	2.24	3.00	3.44	3.62
	mRIC	0.40	3.28	4.76	6.02	6.22
	PIC	0.22	2.20	2.92	3.38	3.52
	FSR	0.10	2.10	2.92	3.38	3.46
	CV	0.44	2.58	3.68	5.76	5.90
	Lasso	2.20	6.94	9.06	9.12	10.32
	RandOracle	0.00	2.04	3.70	4.94	5.94
0.7	AIC	4.28	5.98	7.36	8.58	9.86
	BIC	1.40	3.32	4.84	6.12	7.00
	RIC	0.54	2.32	3.88	4.84	5.34
	mRIC	0.70	4.06	7.40	10.12	11.60
	PIC	0.30	2.28	3.64	4.56	4.58
	FSR	0.20	2.22	3.76	5.14	6.26
	CV	0.34	2.84	4.02	6.86	8.70
	Lasso	3.38	7.34	10.38	12.68	13.62
	RandOracle	0.00	2.00	4.16	6.12	9.84

Table A.4: Average Model Size (n = 50)

ρ	Method	H0	H1	H2	H3	H4
0	AIC	2.76	4.88	6.16	7.88	8.78
	BIC	0.38	2.48	4.44	5.36	6.26
	RIC	0.28	2.26	4.06	5.08	5.74
	mRIC	0.36	3.32	5.64	8.22	10.18
	PIC	0.26	2.22	4.08	5.10	5.90
	FSR	0.16	2.22	4.22	5.32	6.36
	CV	0.64	2.48	5.42	6.80	9.78
	Lasso	1.80	6.02	9.12	10.76	12.50
	RandOracle	0.00	2.00	4.48	6.32	8.14
0.7	AIC	4.00	5.52	7.92	9.64	11.56
	BIC	0.74	2.50	5.10	6.94	8.72
	RIC	0.32	2.28	4.58	6.72	8.26
	mRIC	0.70	3.78	8.44	11.60	14.90
	PIC	0.22	2.22	4.60	6.84	8.46
	FSR	0.10	2.26	4.84	7.30	9.98
	CV	0.76	2.20	5.26	7.86	10.50
	Lasso	2.76	6.72	10.92	13.68	16.18
	RandOracle	0.00	2.00	4.56	7.22	10.18

Table A.5: Average Model Size (n = 150)

ρ	Method	H0	H1	H2	H3	H4
0	AIC	2.14	4.48	6.64	8.80	10.28
	BIC	0.14	2.14	4.38	6.52	7.86
	RIC	0.14	2.16	4.38	6.54	7.90
	mRIC	0.14	3.10	6.36	9.78	12.74
	PIC	0.16	2.14	4.46	6.58	8.00
	FSR	0.08	2.14	4.52	6.88	8.78
	CV	0.24	2.42	4.84	8.70	11.00
	Lasso	1.18	6.30	9.14	11.92	13.94
	RandOracle	0.00	2.00	4.58	7.10	9.82
0.7	AIC	3.00	4.96	8.40	9.92	13.00
	BIC	0.22	2.26	5.50	7.94	10.36
	RIC	0.22	2.26	5.50	7.96	10.36
	mRIC	0.26	3.24	9.30	12.64	16.14
	PIC	0.18	2.24	5.60	8.12	10.70
	FSR	0.08	2.26	5.90	8.54	12.24
	CV	0.20	2.28	6.08	9.02	12.34
	Lasso	1.58	6.10	12.00	14.16	16.64
	RandOracle	0.00	2.00	5.94	8.30	11.30

Table A.6: Average Model Size (n = 500)

Bibliography

- F. Abramovich, Y. Benjamini, D.L. Donoho, and I.M. Johnstone. Adapting to unknown sparsity by controlling the false discovery rate. *Arxiv preprint math/0505374*, 2005.
- H. Akaike. Maximum likelihood identification of Gaussian autoregressive moving average models. *Biometrika*, 60(2):255–265, 1973.
- Y. Benjamini and Y. Gavrilov. A simple forward selection procedure based on false discovery rate control. *eprint arXiv: 0905.2819*, 2009.
- Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.
- J.O. Berger and L.R. Pericchi. The Intrinsic Bayes Factor for Model Selection and Prediction. *Journal of the American Statistical Association*, 91(433), 1996.
- L. Birge and P. Massart. A generalized C_p criterion for Gaussian model selection. *preprint*, 2001.

- S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- L. Breiman. The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *Journal of the American Statistical Association*, pages 738–754, 1992.
- L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- L. Breiman and D. Freedman. How many variables should be entered in a regression equation? *Journal of the American Statistical Association*, pages 131–136, 1983.
- L. Breiman and P. Spector. Submodel selection and evaluation in regression. The X-random case. *International Statistical Review/Revue Internationale de Statistique*, pages 291–319, 1992.
- L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression trees. Wadsworth. Inc., Belmont, CA, 358, 1984.
- E. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n . *Annals of Statistics*, 35(6):2313–2351, 2007.
- H. Chipman, E.I. George, R.E. McCulloch, M. Clyde, D.P. Foster, and R.A. Stine. The practical implementation of Bayesian model selection. *Lecture Notes-Monograph Series*, pages 65–134, 2001.
- T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley-Interscience, 2006.

- D.L. Donoho and J.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- B. Efron and R.J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall, 1997.
- B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of statistics*, pages 407–451, 2004.
- B. Efron, T. Hastie, and R. Tibshirani. Discussion: The Dantzig Selector: Statistical Estimation When p is much larger than n . *Annals of Statistics*, 35(6):2358–2364, 2007.
- D.P. Foster and E.I. George. The risk inflation criterion for multiple regression. *The Annals of Statistics*, pages 1947–1975, 1994.
- DP Foster and RA Stine. Local asymptotic coding and the minimum description length. *IEEE Transactions on Information Theory*, 45(4):1289–1293, 1999.
- D.P. Foster and R.A. Stine. Variable Selection in Data Mining. *Journal of the American Statistical Association*, 99(466):303–313, 2004.
- I.E. Frank and J.H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, pages 109–135, 1993.
- Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156. Citeseer, 1996.

- GM Furnival and R.W. Wilson. Regression by leaps and bounds. *Technometrics*, 16: 499–511, 1974.
- A. Gelman, J.B. Carlin, and H.S. Stern. *Bayesian data analysis*. CRC press, 2004.
- E.I. George. The variable selection problem. *Journal of the American Statistical Association*, pages 1304–1308, 2000.
- E.I. George and D.P. Foster. Calibration and empirical Bayes variable selection. *Biometrika*, 87(4):731–747, 2000.
- E.I. George and R.E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, pages 881–889, 1993.
- D. Harrison and D.L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.
- T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*. Springer New York, 2001.
- T. Hastie, J. Taylor, R. Tibshirani, G. Walther, et al. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1(1):1–29, 2007.
- A.E. Hoerl and R.W. Kennard. Ridge regression: applications to nonorthogonal problems. *Technometrics*, pages 69–82, 1970.
- J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

- C.M. Hurvich and C.L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.
- Ø. Langsrud. Rotation tests. *Statistics and Computing*, 15(1):53–60, 2005.
- E.L. Lehmann, G. Casella, and G. Casella. *Theory of point estimation*. Springer New York, 1998.
- D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univ Pr, 2003.
- D. Madigan, J. York, and D. Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.
- C.L. Mallows. More comments on Cp. *Technometrics*, pages 362–372, 1995.
- P. McCullagh and J.A. Nelder. *Generalized linear models*. Chapman & Hall/CRC, 1989.
- A.J. Miller. *Subset selection in regression*. CRC Press, 2002.
- TJ Mitchell and JJ Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, pages 1023–1032, 1988.
- E.J.G. Pitman. Significance tests which may be applied to samples from any populations: Part II. *Royal Statistical Society Supplement*, 4:225–32, 1937.
- A.E. Raftery, D. Madigan, and J.A. Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, pages 179–191, 1997.

- J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
- G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.
- J. Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, pages 486–494, 1993.
- J. Shao. Bootstrap Model Selection. *Journal of the American Statistical Association*, 91(434), 1996.
- C. Stein and W. James. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 361–379, 1961.
- M. Stone. An asymptotic equivalence of choice of model by cross-validation and Akaike’s criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 44–47, 1977.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- R. Tibshirani and K. Knight. The covariance inflation criterion for adaptive model selection. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 529–546, 1999a.

R. Tibshirani and K. Knight. The covariance inflation criterion for adaptive model selection. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 529–546, 1999b.

Y. Wu, D.D. Boos, and L.A. Stefanski. Controlling variable selection by the addition of pseudovariates. *Journal of the American Statistical Association*, 102(477):235–243, 2007.