Predictions with Many Regressors and Big Data

Julian Herbert Müller
Enrolment Number: 1923031
University of Mannheim

January 10, 2025

Abstract

This seminar paper compares various prediction models within a big data context. The classic OLS Regression is often the first choice for many projects, but applying an OLS Regression indiscriminately to the whole data set is within a big data context often either infeasible due to perfect multicollinearity or leads to low prediction quality due to overfitting/imperfect multicollinearity. The challenge of having too much data to achieve good predictions is a relatively modern problem, and several approaches have been developed to address it. This paper compares some of these approaches, including shrinkage estimators, which will receive particular attention, a principal components based OLS Regression, and specific OLS models that utilize handpicked regressors. The findings show that, in terms of out-of-sample MSPE ($MSPE_{o.o.s}$) minimization, the implemented shrinkage estimators, namely Ridge Regression and LASSO, can achieve small but persistent improvements compared to an OLS Regression based on subject matter considerations or a PCA-based OLS Regression. Notably, this does not translate to per unit of time $MSPE_{o.o.s}$ minimization, which persistently occurs with an OLS Regression that utilizes handpicked regressors. On the other hand, relying on handpicked regressors is prone to subjectivity and uncertainty about optimality. PCA-based OLS Regressions solve these problems and lead to optimal results for a $MSPE_{o.o.s}$ minimization per unit of time if subjectivity has to be eradicated. This indicates that alternative approaches, such as shrinkage estimators or a PCA-based OLS Regression, achieve significant improvements compared to a standard OLS Regression. However, different approaches have distinct strengths and weaknesses.

JEL classification: C53, C55, C52

Keywords: Ridge Regression, LASSO, Prediction, Shrinkage

1 Introduction

In this paper, I explore the issue of "Predictions with Many Regressors and Big Data," focusing primarily on shrinkage estimators. The goal is to examine the theoretical foundations of shrinkage estimators, to apply them to a real dataset, and to compare them with alternative methods. This approach and the selection of estimators implemented are based on Stock and Watson (2020, Chapter 14). Additionally, this will demonstrate the advantages of shrinkage estimators compared to alternative OLS-based methods and provide insights into when these estimators are especially useful or less effective. To be more precise, I implemented Ridge Regression and LASSO as shrinkage estimators, PCA-based OLS Regressions, OLS Regressions based on handpicked variables, and a simple mean. All of these models were implemented in R using no packages. I aimed to maximize prediction quality while predicting the average test scores of schools in California. To measure prediction quality, I used the mean squared prediction error (*MSPE*).

2 Theoretical Foundations

2.1 Norms

After this brief introduction, I would first like to establish some theoretical foundations before discussing the implementation of the various methods and the results in later sections. I begin with the definition of a norm, which is based on Deitmer (2021, Chapter 8, page 183).

Definition 1 Norm *Given a vector space V over $\mathbb{R}$, a norm is a function $\|\cdot\| : V \to \mathbb{R}$ with the following properties for $v, w \in V$:*

1. **Positive Definiteness:** For all $v \in V$, it holds that $\|v\| \geq 0$ and $\|v\| = 0 \iff v = 0$
2. For all $v \in V$ and $\lambda \in \mathbb{R}$, it holds that $\|\lambda v\| = |\lambda| \|v\|$
3. **Triangle Inequality:** For all $v, w \in V$, it holds that $\|v + w\| \leq \|v\| + \|w\|$

A vector space V together with a norm $N(x)$ is called a normed vector space and is denoted by $(V, N(x))$.

This concept will be of particular importance in later sections. It is essential to understand that a norm represents the measure of distance from the origin. Depending on the norm, distances are measured differently, and there are various norms. For the following considerations, a specific type of norm will be particularly relevant, namely the so-called L^p-norms, which have the form:

$$L^p = \left(\sum_{i=1}^{n} |d_i|^p \right)^{\frac{1}{p}}$$

The most prominent L^p-norms are the L_1 norm (also known as the Taxicab or Manhattan norm), given by $\sum_{i=1}^{n} |d_i|$, and the L_2 norm (also known as the Euclidian norm), given by $\sqrt{\sum_{i=1}^{n} d_i^2}$, see Hansen (2022, Chapter 29, page 943). Both will be needed in the following sections.

2.2 Mean Squared Prediction Error (MSPE)

To compare the prediction quality of different models, I use the MSPE. The following definition is based on Stock and Watson (2020, Chapter 14, page 518):

Definition 2 Mean Squared Prediction Error (MSPE) *The (theoretical) mean squared prediction error is the expected value of the square of the prediction error that arises when the model is used to predict an observation not in the data set*

$$MSPE = E((Y_{o.o.s} - \hat{Y}(X_{o.o.s}))^2)$$

An estimator of the MSPE is $\hat{MSPE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

From the perspective of minimizing the MSPE, the best possible prediction is the conditional mean - that is $E(Y_{o.o.s}|X_{o.o.s})$, see Stock and Watson (2020, Chapter 14, page 518). Because of this the MSPE has become a commonly used measure of prediction quality and will also be the tool I primarily use. Being precise in the nomenclature is important here because otherwise confusion is inevitable. When writing MSPE I refer to the expected value $E((Y_{o.o.s} - \hat{Y}(X_{o.o.s}))^2)$, while when writing $MSPE_{i.s}$ I refer to an estimate of the MSPE ($\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$), that is calculated using "pseudo" out-of-sample data, utilizing only the training data set and k-fold Cross Validation. When writing $MSPE_{o.o.s}$ I refer to an estimate of the MSPE as well, but one that is calculated using "real" out-of-sample data, i.e., utilizing the testing data set. The relevance of this distinction will become clearer during the sections 3 and 4, but should be kept in mind throughout the paper.

2.3 Ordniary Least Squares (OLS)

The following definition of the OLS estimator is based on Stock and Watson (2020, Chapter 6, page 221):

Definition 3 Ordinary Least Squares Estimator (OLS) *The estimator* $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$ *for the coefficients of the conditional expectation function* $E(Y|X)$*, where* $E(Y|X)$ *is assumed to be linear in the coefficients and therefore have the form* $E(Y|X) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$*, is referred to as the OLS estimator if it is the solution to the following optimization problem:*

$$\min_{\hat{\beta}} \sum_{i=1}^{n} \hat{u}_i^2 \qquad [1]$$

Where $\sum_{i=1}^{n}\hat{u}_i^2 = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1 - ... - \hat{\beta}_k x_k)^2 = (Y - X\hat{\beta})'(Y - X\hat{\beta})$*. The solution to the optimization problem above is given by* $\beta_{OLS} = (X'X)^{-1}X'Y$*.*

Given this minimization problem, the name of the estimator becomes clear; we find the regression line that minimizes the sum of the squared residuals, where the residuals are defined by $\hat{u}_i = y_i - \hat{y}_i$. This optimization problem is equivalent to the linear projection of $Y = (y_1, ..., y_n)$ onto $X = (X_1, ..., X_k)$, where $X_i = (x_{i,1}, ..., x_{i,n})$, compare Wickens (2014, Chapter 4, page 46). The OLS estimator is the classic estimator, which is implemented in almost all cases.

However, the OLS estimator particularly shows weaknesses in contexts with many regressors. In such situations, it can even completely break down, making it uncomputable. When there are more regressors than observations $X'X$ would no longer be invertible, and thus $\beta_{OLS} = (X'X)^{-1}X'Y$ would no longer be computable (this problem is referred to as perfect multicollinearity). Even if OLS was computable because there are fewer regressors than observations, as the number of regressors approaches the number

of observations, it leads to what is known as imperfect multicollinearity, which is often referred to as overfitting. The OLS estimator tends to overfit the sample data in such a setup, leading to poor out-of-sample prediction quality. Therefore, a simple, thoughtless OLS Regression, which uses all possible variables, can lead to extremely poor prediction quality. Accordingly, one has to turn to alternative approaches when dealing with these setups.

2.4 Shrinkage Estimator

2.4.1 General Principle

The class of shrinkage estimators share a fundamental characteristic: they introduce bias but can thereby reduce variance. This characteristic is often referred to as a "bias-variance tradeoff". As a result, they provide an estimator that, while biased, has lower variance than classical estimators. This can lead to an improved MSE when the variance is high and the bias is relatively low (the MSE of an estimator $\hat{\theta}$ for θ is defined as $MSE(\hat{\theta}) = E((\theta - \hat{\theta})^2) = Var(\hat{\theta}) + (E(\hat{\theta}) - \theta)^2)$. The following example of a simple shrinkage estimator is based on Hansen (2022, Chapter 28, page 884). It can be described as follows:

Example 1 Simple Shrinkage Estimator *For an estimator $\hat{\theta}$ with $Var(\hat{\theta}) = \sigma^2$ and expectation $E(\hat{\theta}) = \theta$, the estimator is defined as*

$$\tilde{\theta} = (1-w)\hat{\theta},\ w \in (0,1)$$

For this estimator, it holds that $Var(\tilde{\theta}) = (1-w)^2\sigma^2$ and $E(\tilde{\theta}) = (1-w)\theta$. Thus, we have $MSE(\hat{\theta}) = \sigma^2$ and $MSE(\tilde{\theta}) = (1-w)^2\sigma^2 + (w\theta)^2$

We can improve the MSE relative to the unbiased estimator when θ is close to 0 and σ^2 is large. A generalization of this idea leads to the so-called Stein shrinkage estimator, see Hansen (2022, Chapter 28, page 885). It should be noted here that, as prediction isn't a concern in this setup, the MSE has been used. Nonetheless this example illustrates the general principle of shrinkage estimators and assuming that $(X_{o.o.s}, Y_{o.o.s})$ are randomly drawn from the same population distribution as the estimation sample, a similar argument could be used utilizing the MSPE and estimating $E((Y_{o.o.s} - \hat{Y}(X_{o.o.s}))^2)$, see Stock and Watson (2020, Chapter 14, page 519).

2.4.2 Ridge Regression

The so-called Ridge Regression is the first shrinkage estimator that I will examine in detail. The following definition is based on Stock and Watson (2020, Chapter 14, page 524):

Definition 4 Ridge Regression *The estimator $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$ for the coefficients of the conditional expectation function $E(Y|X)$, where $E(Y|X)$ is assumed to be linear in the coefficients and therefore have the form $E(Y|X) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$, is referred to as Ridge Regression if it is the solution to the following optimization problem:*

$$\min_{\hat{\beta}} \sum_{i=1}^{n} \hat{u}_i^2 + \lambda \sum_{j=1}^{k} \hat{\beta}_j^2 \qquad [2]$$

Where $\sum_{i=1}^n \hat{u}_i^2 + \lambda \sum_{j=1}^k \hat{\beta}_j^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1 - ... - \hat{\beta}_k x_k)^2 + \lambda \sum_{j=1}^k \hat{\beta}_j^2 = (Y - X\hat{\beta})'(Y - X\hat{\beta}) + \lambda \sum_{j=1}^k \hat{\beta}_j^2$. The solution to the optimization problem above is given by $\beta_{Ridge} = (X'X + \lambda I)^{-1} X'Y$.

This estimator can be viewed from two perspectives. On the one hand, as an ordniary optimization

problem, while OLS minimizes [1], Ridge Regression minimizes [2]. It introduces a "penalty term", $\lambda \sum_{j=1}^{k} \hat{\beta}_j^2$, which inflates if single $\hat{\beta}_j$s increase. On the other hand, it can also be understood as a constrained optimization problem, where λ is the Lagrange parameter. Thus, we optimize $\sum_{i=1}^{n} \hat{u}_i^2$ under the constraint $\sum_{j=1}^{k} \hat{\beta}_j^2 = \tau$. Therefore the Lagrange function is given by $\mathscr{L}(\hat{\beta}, \lambda) = \sum_{i=1}^{n} \hat{u}_i^2 + \lambda \left(\sum_{j=1}^{k} \hat{\beta}_j^2 - \tau \right)$, this is equivalent to optimizing [2], see Hansen (2022, Chapter 29, page 918). It is important to note that there is a unique relationship between τ and λ, which can be expressed as

$$\tau = Y'X(X'X + \lambda I_p)^{-1}(X'X + \lambda I_p)^{-1}X'Y. \quad [3]$$

Thus, any λ can be translated into a specific constraint under which one optimizes since every τ represents a constraint to the length of β_{Ridge}: $||\beta_{Ridge}|| = \sqrt{\tau} = \sqrt{\sum_{j=1}^{k} \beta_{Ridge,j}^2}$, see Hansen (2022, Chapter 29, page 945). It is also important to remember that we use the Euclidean norm, or L_2 norm, to formulate this constraint. The rationale behind using this optimization problem is the utilization of a "bias-variance tradeoff". By introducing the constraint, the solution lies closer to zero (with a sufficiently strict constraint, i.e., a small τ), thus introducing a bias (distortion of the previously unbiased OLS estimator). However, the variance of this estimator may be sufficiently smaller than that of the OLS estimator, resulting in an overall reduction in MSPE. Furthermore this estimator can always be implemented even if there are more regressors than observations, this is because $X'X + \lambda I$ can be inverted even if $X'X$ is singular.

2.4.3 Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO is the next shrinkage estimator I would like to examine more closely. The following definition is based on Stock and Watson (2020, Chapter 14, page 528)

Definition 5 Least Absolute Shrinkage and Selection Operator (LASSO) *The estimator $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$ for the coefficients of the conditional expectation function $E(Y|X)$, where $E(Y|X)$ is assumed to be linear in the coefficients and therefore have the form $E(Y|X) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$, is referred to as LASSO if it is the solution to the following optimization problem:*

$$\min_{\hat{\beta}} \sum_{i=1}^{n} \hat{u}_i^2 + \lambda \sum_{j=1}^{k} |\hat{\beta}_j| \quad [4]$$

Where $\sum_{i=1}^{n} \hat{u}_i^2 + \lambda \sum_{j=1}^{k} |\hat{\beta}_j| = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1 - \ldots - \hat{\beta}_k x_k)^2 + \lambda \sum_{j=1}^{k} |\hat{\beta}_j| = (Y - X\hat{\beta})'(Y - X\hat{\beta}) + \lambda \sum_{j=1}^{k} |\hat{\beta}_j|$

Similar to Ridge Regression, LASSO modifies the optimization problem for calculating the beta vector by introducing a "penalty term". This can be formulated as a Lagrange optimization problem with the Lagrange function $\mathscr{L}(\hat{\beta}, \lambda) = \sum_{i=1}^{n} \hat{u}_i^2 + \lambda \left(\sum_{j=1}^{k} |\hat{\beta}_j| - \tau \right)$, which is equivalent to optimizing [4], see Hansen (2022, Chapter 29, page 918). It is noteworthy that the only change we make relative to Ridge Regression is the norm used to measure the length of the beta vector. Once again, we limit the length of the beta vector, but this time the length of the beta vector is measured using the Taxicab/L_1 norm. The second part of the name of LASSO is due to a specific characteristic of LASSO. It utilizes the L_1 norm, which leads to "not smooth" constraints, and this again leads to the coefficients either being zero or comparatively large, this characteristic is often referred to as "variable selection". Therefore, some argue that the interpretability of LASSO is better compared to Ridge Regression, see James, Witten, Hastie and

Tibshirani (2021 Chapter 6, page 242). But the aim of this paper is maximization of prediction quality, not interpretability, therefore these concerns aren't further discussed in this paper.

2.5 Principal Component Analysis (PCA)

In contrast to the previously discussed Ridge Regression and LASSO, a PCA-based OLS Regression takes a fundamentally different approach. The problems originally encountered with the OLS estimator were perfect or imperfect multicollinearity. One approach therefore is to retain as much information as possible while reducing the number of regressors. We can understand information as the "variance/covariance structure" of the data. Therefore, the aim is to capture as much variance of the original dataset as possible with as few regressors as possible. This is done while no longer necessarily using the original regressors but allowing any normed linear combination to serve as regressors under the constraint that these linear combinations are orthogonal to each other. These new regressors are then called principal components (PC). Then, the first k PCs are used for an OLS Regression as regressors, see Stock and Watson (2020, Chapter 14, page 532). This naturally reduces the issue of imperfect multicollinearity as only a (k $<$ n)-dimensional subspace is spanned, and the data vectors are now orthogonal to each other. The following theorem is based on Johnson and Wichern (2014, Chapter 8, page 432):

Theorem 1 Principal Component Analysis (PCA) *Let Σ be the covariance matrix associated with the random vector $\mathbf{X}^T = (X_1, X_2, \ldots, X_p)$. Let Σ have the eigenvalue-eigenvector pairs $(\lambda_1, \mathbf{e}_1), (\lambda_2, \mathbf{e}_2), \ldots, (\lambda_p, \mathbf{e}_p)$, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$. Then the j-th population principal component is given by*

$$Y_j = \mathbf{e}_j^T \mathbf{X} = e_{j1}X_1 + e_{j2}X_2 + \cdots + e_{jp}X_p, \quad j = 1, \ldots, p.$$

Thereby Y_j is the linear combination, with the weighting vector having unit length, of $\boldsymbol{X}$ with the highest possible variance, under the constraint that Y_j has to be orthogonal to $(Y_1, ..., Y_{j-1})$

3 Implementation

3.1 Packages

Even so there are several packages available for implementing the shrinkage estimator, the most prominent one being `glmnet`, I implemented everything from scratch, without any packages. This allows me to fully decide how to implement these models. This problem of control is often overlooked, for example `glmnet` uses a scaled optimisation problem:

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} R_\lambda(\beta_0,\beta) = \min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} \left[\frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - x_i^\top \beta \right)^2 + \lambda \sum_{j=1}^{p} \left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right] \right]$$

for estimating LASSO and Ridge Regression, as extreme cases of Elastic Net, compare Friedman, Hastie, and Tibshirani (2010). This deviates from [2] and [4]. For similar reasons, I also decided to use no packages for other applications implemented in this paper.

3.2 Training and Testing

3.2.1 Training Data Set and Testing Data Set

Initially, I randomly divided the data set into two equally large subsets, a "training data set" and a "testing data set". Most calculations took place on the training data set. This includes the estimation of the coefficients, the k-fold Cross Validation, and the optimal λ estimation, all topics I will further comment on in the following sections. Solely the calculation of the $MSPE_{o.o.s}$, which is used to compare the prediction quality of the different models, took place on the testing data set, using the coefficient vectors estimated on the training data set, based on the optimal λ, estimated on the training data set.

3.2.2 Optimal λ Algorithm (1)

When it comes to implementing alternative prediction models besides OLS, the main difficulty usually is not the implementation of the estimator, as it is available in closed form for Ridge Regression and PCA-based OLS Regressions. Rather, the more complicated question is how the externally specified parameters should be set. For Ridge Regression and LASSO, the issue arises regarding the constraint under which the optimization should be conducted. For this "optimal λ estimation" I implemented two algorithms:

The first algorithm for the optimal λ estimation is a kind of "grid search" (this algorithm will be referred to as (1) during the rest of this paper). I calculate an in-sample MSPE ($MSPE_{i.s}$) for λs between 1 and 100,001. The step size chosen depends on the computational power of the computer; I decided on a step size of 20,000, resulting in a total of 5 $MSPE_{i.s}$s calculated for 5 different constraints. After this is done, I select the λ that resulted in the lowest $MSPE_{i.s}$, as well as the λs immediately to the right and left of this λ_{min}. Next, I calculate λs within this interval using a step size of 10,000 and compare their $MSPE_{i.s}$s, choosing the λ with the lowest $MSPE_{i.s}$ from this set. This process continues with the interval stretched from the λ before to the λ after λ_{min} of the specific stage, for the step sizes of 5000, 2500, 1250, 625, 125, 25, 5, and finally 1. Ultimately, one obtains the λ from the natural numbers that has the lowest $MSPE_{i.s}$, provided that the $MSPE_{i.s}$ values exhibit a quadratic structure as a function of λ, which is usually the case, see Stock and Watson (2020, Chapter 14, page 526).

3.2.3 Optimal λ Algorithm (2)

The second variant I implemented is based on Kascha and Trenkler (2015) (this algorithm will be referred to as (2) during the rest of this paper). The idea is somewhat different from (1). Instead of simply generating λs over a large range of potential values and then zooming in on the "promising" λs, this approach first narrows down the set of potential constraints based on $||\beta_{OLS}||$. Then, λs are calculated on a logarithmic scale, allowing for more λs for small values and fewer for large values. The optimal λ, in the sense of $MSPE_{i.s}$ minimization, from this set is then used. More precisely formulated, we can choose α such that:

$$||\beta_{Ridge}|| = \sqrt{\tau} = \alpha \sqrt{\sum_{j=1}^{k} \beta_{OLS,j}^2} = \alpha ||\beta_{OLS}|| \iff \tau = (\alpha ||\beta_{OLS}||)^2$$

Since [3] establishes a connection between τ and λ, we can derive the constraint relative to $||\beta_{OLS}||$ and thus conclude on λ_{max} and now calculate λs on a logarithmic scale between 0 and λ_{max}. I decided to

choose 0.01% of $||\beta_{OLS}||$ as $\sqrt{\tau}$ to ensure a sufficiently strict constraint.

3.2.4 Further Considerations and Usage of (1) and (2)

The specific setups will be discussed in more detail in section 4.1, but there are a few important considerations regarding the setups when it comes to implementation. First of all it is important to note that (2) assumes that OLS is potentially computable, which is only the case for the 1726 regressors setup. Therefore, I computed the τ for 1726 regressors and used this τ as a guideline for implementation of (2) in cases where OLS couldn't be computed due to perfect multicollinearity. It should be noted that (1) has been used to implement Ridge Regression in the case of 2095 regressors, while (2) has been used to implement Ridge Regression in the case of 1726, 2095, 2215, and 2515 regressors and LASSO in the case of 2095 regressors, this is due to the simpler implementation of (2) and its more adjustable computational cost.

3.2.5 k-fold Cross Validation

Essential for the implementation of the various methods is the estimation of the $MSPE_{i.s}$s, as they are utilized by (1) and (2) to estimate the optimal λ. Therefore, I utilized k-fold Cross Validation, but there are alternative approaches using information criteria, which I have not implemented, see Kascha and Trenkler (2015). The idea behind k-fold Cross Validation is to divide the dataset into k random parts of the same size, then combine $k-1$ datasets, which work as a pseudo training data set. Dividing the training data set into a pseudo training data set (k-1 parts) and a pseudo testing data set (kth part) is how an estimation of the MSPE ($MSPE_{i.s,k}$) solely with the training data set is accomplished. To be a bit more precise, for each k, the $MSPE_{i.s,k}$ is computed, and the mean of the $MSPE_{i.s,k}$s is the $MSPE_{i.s}$ for a specific λ. This serves as an estimate for the MSPE of the specification. This estimation procedure should ensure that the selected configuration performs well across different data splits. It's common to choose k = 10, this has been shown to be a good compromise between computational demand and accuarcy, therefore I decided to use k = 10 as well, see Hansen (2022, Chapter 29, page 869).

3.3 Ridge Regression Implementation

The implementation of β_{Ridge} is comparatively simple, once one has decided which algorithm to use. As β_{Ridge} has a closed-form solution, it is possible to just compute this estimator using matrix operations in R. When implementing (2), it is important to decide how many λs to calculate. Based on Kascha and Trenkler (2015) I decided to calculate 100 λs on the logarithmic scale, for the Ridge Regression applications.

3.4 LASSO Implementation

The estimation of the optimal λ for LASSO follows similar considerations as the estimation of the optimal λ for Ridge Regression. However, LASSO was implemented only using method (2) due to the significant computational effort associated with LASSO. Additionally, I had to reduce the number of λs computed from the normally implemented 100 to 6. Another challenge was the implementation of LASSO itself. While the Ridge Regression coefficient vector has a closed-form solution, this is no longer the case for LASSO due to the use of the L_1 norm. Therefore, LASSO must be numerically approximated without inducing too much computational effort. I accomplished this using the 'optim' function

in R, which numerically computes the optimum of a function. Once this is done, the implementation is fundamentally the same as for Ridge Regression.

3.5 PCA Implementation

The estimation of the optimal number of Principal Components (PCs) is heavily based on (1), but this time, the maximum number of possible PCs is known (equal to the number of regressors). Additionally, the result must come from the natural numbers, making (1) particularly suitable for estimation. Accordingly, I implemented (1) with step sizes of 500, 50, 10, and 1. Calculating the optimal number of PCs simply builds upon the methods used for implementing Ridge Regression. However, alternative methods for determining the optimal number of PCs exist, like utilizing a scree plot or the percentage of explained total variance. Nonetheless, I decided to use (1), as it allows for a decision on the number of PCs to use without any subjective judgment or rule of thumb that might be suboptimal for the specific data set.

3.6 Percentile-Bootstrap Confidence Intervals (PBCI)

Using the methods described so far only a point estimation would be possible for the different models and metrics. This is insufficent as it easily becomes unclear if an $MSPE_{o.o.s}$ improvement is structural or only due to randomness. Therefore I implemented PBCIs, they are constructed for an estimator T_n for θ by generating so-called "bootstrap samples" of the original data set. This is accomplished by randomly drawing (with replacement) observations from the n original observations n times. This process is repeated N (in my case 100) times so that now N new n-dimensional samples exist, which were constructed by randomly drawing observations from the original data set. On each of these N bootstrap samples, the estimator T_n is computed ($T_{n,i}*$). If this is done, the interval [5%-quantile($T_{n,i}*$), 95%-quantile($T_{n,i}*$)] is the asymptotic 90% confidence interval (CI) for the estimator T_n, if a monotone transformation $U = m(T_n)$ exists such that U $\sim N(\phi, c^2)$, where $\phi = m(\theta)$, this approach is based on Wasserman (2006, Chapter 3, page 34). Using these PBCIs I estimated the 90% CIs for the various Ridge Regression and OLS setups, implementation for a PCA-based OLS Regression or LASSO wasn't possible due to the computational demands of these methods. Additionally, it is very important to stress that it is unclear whether the mathematical assumptions are rigorously fulfilled in my case. Nonetheless, the estimated CIs should at least give a rough idea of the prediction quality one should expect and allow to differentiate more clearly between random differences and structural differences.

4 Empirical Prediction Comparison

4.1 Data and Setup

The dataset I worked with is the `ca_school_testscores` dataset. The full dataset consists of data gathered on 3932 elementary schools in the state of California in 2013. The dependent variable I examined is a linear combination of Mathematics scores and English/Language-Arts scores, defined as follows:

$$score_i = \alpha \cdot math_i + (1 - \alpha) \cdot elarts_i$$

This allows for weighing how much emphasis I want to place on each score. Stock and Watson decided to use $\alpha = 0.5$, see Stock and Watson (2020, Chapter 14, page 517). The data exists at the school, district, and zip code levels, and there are 65 base variables, 5 of which are binary. Additionally, it should be noted that all regressors were standardized, and the dependent variables were mean-centered to ensure

comparability, see Stock and Watson (2020, Chapter 14, page 519). Because I divided my original data set into a training and a testing data set, it's infeasible to compute OLS for any configuration that includes more than 1966 regressors. My basic model includes 2095 variables, generated by squaring, cubing, taking the natural logarithm, and creating interaction terms, therefore OLS can't be computed for this setup. A second configuration includes only 1726 variables, derived by modifying the base variables (excluding some less "meaningful" ones) and removing the natural logarithm as a transformation. In this second configuration, an OLS Regression is possible, but it suffers from high imperfect multicollinearity. Therefore, I can directly compare OLS with Ridge Regression using the same regressors and dependent variable. A third configuration adds the 4th and 5th powers to the base configuration, which leads to 2215 regressors. Furthermore a fourth configuration implements additional nonlinear transformations of the base variables, including all powers up to the 10th power, this leads to 2515 regressors. My comparison instrument between the various models is the $MSPE_{o.o.s}$.

4.2 Results

4.2.1 General Comments

In the following, I will compare the different models and approaches on various levels. I will use the $MSPE_{o.o.s}$ as a measure for the prediction quality. Table 1 and Table 2 provide a compact summary of the results of my work. All data and results referenced can also be found in the RMarkdown output and Code provided with this paper.

Table 1: **MSPE and Absolute/Relative Improvement**

Model	$MSPE_{o.o.s}^{100\%}$	$MSPE_{o.o.s}^{75\%}$	$MSPE_{o.o.s}^{50\%}$	$MSPE_{o.o.s}^{25\%}$	$MSPE_{o.o.s}^{0\%}$	$\frac{MSPE_{o.o.s}}{MSPE_{Avg}}$	$\frac{\Delta_{MSPE_{o.o.s},Avg}}{Time}$
OLS 1726	1036993.0000	295545.0000	187743.6000	49787.3200	130819.8000		
Average	1720.9700	1294.4470	982.5172	761.9378	557.9097	100.00%	0
Ridge 1726	806.8532	574.7741	390.1871	219.1890	152.6194	37.42%	0.1095
Ridge 2095 (1)	811.1861	587.0026	372.9812	248.0774	151.8341	38.04%	0.0459
Ridge 2095 (2)	832.9751	563.2653	384.5102	240.6322	144.5796	37.71%	0.0350
Ridge 2215	822.4967	583.7683	392.0700	230.1321	150.7986	38.01%	0.0300
Ridge 2515	828.7387	577.8676	376.2091	244.9363	154.0259	38.17%	0.0211
LASSO 2095	837.2508	626.3619	385.9010	239.3404	151.8766	39.00%	0.0092
PCA 2095	1176.9550	981.2556	619.0997	536.5463	271.6690	65.26%	0.7478
OLS1	1689.9000	1273.2200	972.0042	747.8535	553.8317	98.58%	16.1945
OLS2	1294.4970	976.8594	694.8502	483.1914	321.6527	68.50%	309.3462
OLS3	923.0036	664.9383	434.3240	276.5282	167.2922	43.10%	570.3391
OLS4	879.2932	613.7437	405.5047	257.2242	155.3513	40.28%	601.3329
OLS5	871.5299	614.4588	404.8539	255.8196	156.9732	40.21%	602.8293
OLS6	867.8640	608.3939	406.0448	255.2433	156.8461	40.07%	604.6779
OLS7	862.9454	603.0430	404.2722	251.3065	153.8380	39.69%	608.4753
OLS8	863.7931	604.4703	400.8004	250.2476	153.2851	39.60%	609.0370
OLS9	864.6583	608.7980	401.6793	249.9050	153.1618	39.68%	607.9159
OLS10	865.4471	608.1719	400.9014	249.1077	152.8764	39.63%	608.2554
OLS11	874.0754	606.1539	399.3613	247.8264	152.3697	39.62%	607.5990

Note: The R command system.time$elapsed + 1 is used to measure computation time. The values shown are based on the average $MSPE_{o.o.s}$ improvement over all 5 dependent variables. The percentage numbers refer to the proportion of the Mathematics scores.

4.2.2 Variance Differences

It is immediately apparent that there are enormous differences in $MSPE_{o.o.s}$ values across the dependent variables. The $MSPE_{o.o.s}$ values for the pure English scores amount to on average 19.85% of the $MSPE_{o.o.s}$ scores for the pure Mathematics scores. This is primarily explained by the variance of the English and Mathematics scores. It turns out that $Var_{Mathematics} = 1711.173$ and $Var_{English} = 581.6659$, making the variance of the English scores only about 33.99% of the variance of the Mathematics scores.

Table 2: **MSPE-Ratio 90% Confidence Intervals**

Model	$\frac{MSPE^{100\%}_{o.o.s}}{MSPE_{Avg}}$	$\frac{MSPE^{75\%}_{o.o.s}}{MSPE_{Avg}}$	$\frac{MSPE^{50\%}_{o.o.s}}{MSPE_{Avg}}$	$\frac{MSPE^{25\%}_{o.o.s}}{MSPE_{Avg}}$	$\frac{MSPE^{0\%}_{o.o.s}}{MSPE_{Avg}}$	overall $\frac{MSPE_{o.o.s}}{MSPE_{Avg}}$
Ridge 1726	[0.4260,0.4806]	[0.3865,0.4386]	[0.3351,0.3809]	[0.2747,0.3120]	[0.2373 0.2690]	[0.3323,0.3753]
Ridge 2095 (1)	[0.4281,0.4783]	[0.3967,0.4438]	[0.3436,0.3864]	[0.2717,0.3057]	[0.2358,0.2649]	[0.3362,0.3753]
Ridge 2095 (2)	[0.4333,0.4855]	[0.3945,0.4407]	[0.3428,0.3854]	[0.2703,0.3046]	[0.2393,0.2681]	[0.3367,0.3763]
Ridge 2215	[0.4340,0.4817]	[0.3964,0.4406]	[0.3438,0.3830]	[0.2739,0.3069]	[0.2360,0.2679]	[0.3383,0.3749]
Ridge 2515	[0.4367,0.4867]	[0.4030,0.4482]	[0.3458,0.3841]	[0.2787,0.3067]	[0.2417,0.2653]	[0.3416,0.3778]
OLS1	[0.9332,1.0307]	[0.9581,1.0554]	[0.9602,1.0570]	[0.9294,1.0296]	[0.9634,1.0830]	[0.9544,1.0483]
OLS2	[0.7232,0.8056]	[0.7078,0.7907]	[0.6667,0.7446]	[0.6022,0.6707]	[0.5771,0.6507]	[0.6587,0.7315]
OLS3	[0.5115,0.5749]	[0.4767,0.5346]	[0.4173,0.4650]	[0.3393,0.3767]	[0.2991,0.3326]	[0.4112,0.4546]
OLS4	[0.4734,0.5364]	[0.4378,0.4959]	[0.3833,0.4304]	[0.3125,0.3499]	[0.2770,0.3105]	[0.3790,0.4236]
OLS5	[0.4716,0.5337]	[0.4385,0.4936]	[0.3844,0.4292]	[0.3112,0.3493]	[0.2761,0.3107]	[0.3798,0.4209]
OLS6	[0.4671,0.5311]	[0.4310,0.4920]	[0.3762,0.4297]	[0.3051,0.3462]	[0.2735,0.3112]	[0.3723,0.4209]
OLS7	[0.4655,0.5268]	[0.4300,0.4890]	[0.3745,0.4248]	[0.3057,0.3431]	[0.2701,0.3041]	[0.3705,0.4161]
OLS8	[0.4633,0.5300]	[0.4277,0.4886]	[0.3744,0.4227]	[0.3024,0.3425]	[0.2660,0.3016]	[0.3701,0.4150]
OLS9	[0.4617,0.5271]	[0.4259,0.4881]	[0.3726,0.4215]	[0.2993,0.3400]	[0.2633,0.2994]	[0.3686,0.4135]
OLS10	[0.4625,0.5276]	[0.4268,0.4869]	[0.3730,0.4206]	[0.2983,0.3380]	[0.2617,0.3001]	[0.3687,0.4130]
OLS11	[0.4620,0.5285]	[0.4258,0.4877]	[0.3711,0.4208]	[0.2970,0.3367]	[0.2622,0.3010]	[0.3669,0.4135]

Note: The CIs shown are PBCIs, as discussed in section 3.6, N = 100 and the percentage numbers refer to the proportion of the Mathematics scores.

Accordingly, English scores are easier to predict than Mathematics scores. However, it is not entirely clear whether this is the only reason for the difference. It might be that English scores are not only easier to predict due to their lower variance but also because the dispersion in English scores may be structurally easier to predict with the observed variables compared to Mathematics scores. The data pertain to the state of California, with information available at the district, zipcode, and school levels. Moreover, data on foreign students, English learners, income, and ethnic diversity is available. Perhaps these kind of variables are especially potent for predicting English scores rather than Mathematics scores.

4.2.3 OLS 1726

Aside from this general finding regarding the nature of the results, it is also striking how a simple OLS Regression collapses when applied indiscriminately to all data due to the problems discussed in 2.3. For the 1726 regressors setup the $MSPE_{o.o.s}$ of the Ridge Regression is on average about 0.21% of the $MSPE_{o.o.s}$ of the simple OLS Regression. This indicates a 99.79% reduction in $MSPE_{o.o.s}$. While this outcome was expected, it underscores how problematic the unconsidered use of OLS Regression can be. It also highlights that alternative estimators, such as shrinkage estimators, can indeed fulfill their purpose and massively reduce the estimator's variance, leading to a reduction in $MSPE_{o.o.s}$ despite the introduction of bias.

4.2.4 Ridge Regression: Further Nonlinearities and Alternative Algorithms

However, the question arises regarding the extent to which the introduction of more nonlinear transformations improves the predictive quality of a shrinkage estimator, here Ridge Regression. Figure 1 shows the estimated 90% CIs for the various Ridge Regression setups, it can be clearly seen, that there is no relevant difference between the outcomes of the different setups. This is not surprising given that even the "smallest" model accesses 1726 different variables. Thus, the introduction of additional variables, constructed from the original 65 base variables, does not yield meaningful added value. Furthermore it can also be seen that there is no relevant difference between the outcomes of (1) compared to (2). If one takes the average of the differences of the on the bootstrap sample estimated MSPE-Ratios between (1) and (2), the result is -0.093% points, which isn't relevant. This implicates that both algorithms lead to very similar results and can be used interchangeably, at least for this data set and model setup.

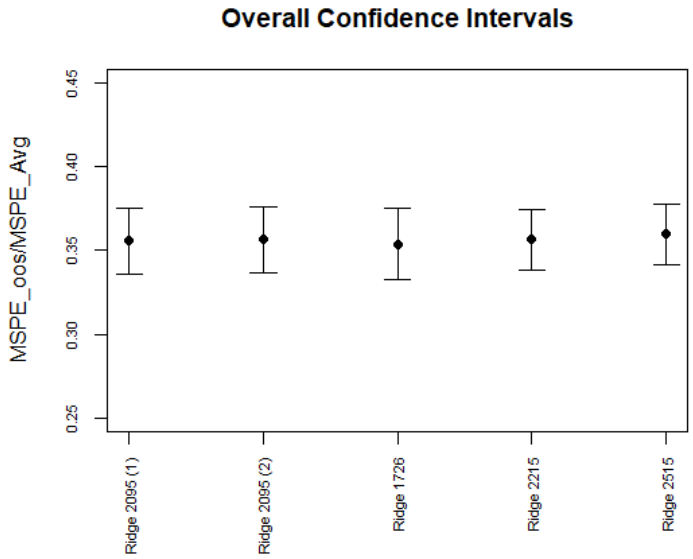

Figure 1: **90% Confidence Intervals, Ridge Regression**

4.2.5 Alternative Models: Average

Another important question is whether simpler less computationally expensive methods or alternative shrinkage estimator lead to similar results or even outperform Ride Regression. I will proceed from the simplest to the more complex methods and compare the results relative to those of Ridge Regression. I began with the simplest possible approach, namely using a simple mean. I took the average of the average test scores of the schools and used that as a naive estimator for the means of the schools not in the training dataset, i.e., those in the testing dataset. This simple estimator achieves an average $MSPE_{o.o.s}$ of about 0.62% of the $MSPE_{o.o.s}$ of the simple OLS Regression with 1726 variables. A significant $MSPE_{o.o.s}$ improvement through a very simple alternative estimator has been achieved. It illustrates that simpler models can yield better results, especially in a context with many regressors and big data, where OLS can quickly lead to detrimental overfitting. Nevertheless, it is evident that the average still performs significantly worse than the Ridge Regression with 1726 variables. The $MSPE_{o.o.s}$ of the Ridge Regression with 1726 variables is on average about 37.42% of the $MSPE_{o.o.s}$ of the average ($MSPE_{Avg}$). Thus, Ridge Regression can again consistently deliver strong $MSPE_{o.o.s}$ improvements. This is good news, as complex models often only lead to marginal improvements relative to a simple mean. But it should be noted that per second of additional computation time Ridge Regression 1726 only leads to a 0.1095 decrease in $MSPE_{o.o.s}$, relative to using the average.

4.2.6 Alternative Models: OLS Regression with Handpicked Regressors

The next step in the complexity hierarchy of the implemented models is the use of OLS models that only utilize a subset of all variables. OLS tends to suffer from the problems described in section 2.3. Fortunately one is not obliged to pass all variables to the OLS Regression. Therefore I provided the OLS Regression only with variables or groups of variables that I believe should have a particularly high explanatory power. I start with one variable and gradually add more variables/groups of variables. The resulting models are called OLS1 to OLS11. Table 3 summarizes the selected groups of variables and the corresponding regressors. The first variable I introduced is the student-teacher ratio. In fact, a simple OLS Regression that includes only the student-teacher ratio achieves $MSPE_{o.o.s}$ values averaging

98.58% of $MSPE_{Avg}$. This is a very similar result to using the average. Next, I introduced the variables average years of teaching, instruction per student expenditure, median income, and ethnicity diversity index, as a second set of important general characteristics. This OLS2 model achieves $MSPE_{o.o.s}$ values averaging 68.5% of $MSPE_{Avg}$. The introduction of these additional variables, which measure income, teacher experience, and to some extent migration, results in a strong reduction in $MSPE_{o.o.s}$. Next, I add the variables free or reduced meals, English learner, free meals, enrollment, and English language proficient. These variables are somewhat more specific than the previously introduced group but mainly measure English proficiency, income, and migration (or correlate with these factors, therefore a third "general characteristics group").

Table 3: **Variable Groups and Corresponding Regressors for the OLS Specifications**

Groups	**Variables**	**1**	**2**
1. General Characteristics 1	Student-teacher ratio	1	11
2. General Characteristics 2	Average years of teaching, instruction per student expenditure, median income, ethnicity diversity index	2	5
3. General Characteristics 3	Free or reduced meals, English learner fraction, free meals fraction, enrollment, English language proficient	3	6
4. Ethnicity Fractions	Fraction of American Indian, Asian, Black, Filipino, Hispanic, Hawaiian, two or more ethnicities, and not reported ethnicity	4	4
5. Teachers	Number of teachers, fraction of first-year teachers, fraction of second-year teachers, part-time measure	5	7
6. General Expenditures per Student	Expenditure on instructional services, pupil services, ancillary services, community services, enterprise expenditures, general administration, and plant services	6	8
7. Expenditure on Capital/Salaries per Student	Certificated salaries, classified salaries, employee benefits, books and supplies, services & other OP expenditures	7	9
8. Age Fractions	Fraction of 5 to 17, 18 to 24, 25 to 34, 35 to 44, 45 to 54, and 55 to 64 year olds	8	10
9. Population	Total population one year or older, fraction of male population, fraction now married, fraction now divorced, fraction now widowed	9	3
10. Education	Fraction of people with a high school diploma, some college or AA, bachelor's degree, and graduate or professional degree and fraction of housing owner	10	1
11. Migration and Unified	Fraction of people moved in from the same county, different county, different state, from abroad, and whether the district is unified	11	2

Note: 1 and 2 refer to the ordering at which the different groups were added. 1 is, therefore the original ordering of adding first the first group, then the second, and so on. 2 is an alternative order of adding the groups to the OLS Regression

This OLS3 model achieves $MSPE_{o.o.s}$ values averaging 43.1% of $MSPE_{Avg}$. Once again significant $MSPE_{o.o.s}$ improvements have been achieved through the introduction of new variables. From now on I add more "specific" variable groups. I started adding "specific" variable groups, as the first 3 groups reflect the variables I assumed to be particularly important; after adding them, I resorted to gradually adding additional variables for specific characteristics. Initially, I achieved large $MSPE_{o.o.s}$ improvements through the introduction of additional variables, but after adding the first three general characterisitcs groups, the $MSPE_{o.o.s}$ improvements reduce drastically. OLS8 achieves $MSPE_{o.o.s}$ values that average approximately 39.60% of $MSPE_{Avg}$. Notably, this represents the minimum of the average $MSPE_{o.o.s}$ values relative to the $MSPE_{Avg}$ values. The introduction of further variables and variable groups only leads to increases or stagnation in the $MSPE_{o.o.s}$ values. OLS8 achieves a 609.037 $MSPE_{o.o.s}$ reduction per second relative to using the average. Even so the fully linear OLS models with handpicked regressors don't reach the $MSPE_{o.o.s}$ levels of Ridge Regression, they significantly outperform Ridge Regression from a $MSPE_{o.o.s}$ reduction per unit of additional computation time perspective. Nonetheless it has to be emphasized that, especially when computing CIs for the OLS Regressions and the Ridge Regressions and comparing them, it becomes clear that Ridge Regression structurally outperforms the OLS Regression, when it comes to an absolute $MSPE_{o.o.s}$ reduction, as can be seen in Figure 2.

Another extremely important point here is that even so OLS8 is the optimal linear OLS model from an overall perspective, OLS7 is optimal for predicting 100% and 75% Mathematics score, while OLS11 optimal for predicting 50%, 25% and 0% Mathematics scores. It is not clear why this is exactly the case. It seems very reasonable to assume that this might be due to the apriori structuring of the variable groups. Changing the order of adding the groups or changing the makeup of the different groups will surely lead to other outcomes. In this case, specifically, it might be the case that group 11, "Migration and Unified," as well as group 10, "Education," are especially important for predicting English scores and less important for predicting Mathematics scores, which leads to the difference in the optimal model setup, based on the dependent variable. Allowing for a different ordering, like I did with 2, emphasizes this point. For 2 the $MSPE_{o.o.s}$ development over the different setups is very different, it turns out that adding group 4 "Ethnicity Fractions" and group 3 "General Characteristics 3" explains the most, furthermore the new "OLS11" now also leads to the lowest $MSPE_{o.o.s}$ for 75% Mathematics scores. Additionally, if one decides on an alternative grouping of the variables based on the distances or correlations between the variables, the results also change drastically. When utilizing the L_1 norm for example setup 9 is optimal for 100% Mathematics score, setup 8 is optimal for 75% Mathematics score, setup 10 is optimal for 50% and setup 7 is optimal for 25% and 0% Mathematics score. These points emphasize the biggest issues with this solution to the problems described in section 2.3. It is unclear what setup is optimal, subjectivity can't be eradicated and domain knowledge is necessary, as well as the possibility to interpret the different regressors and their meaning.

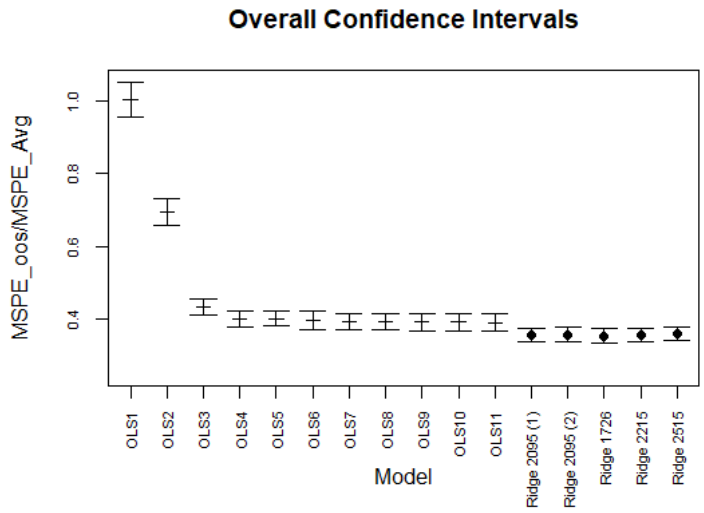

Figure 2: **90% Confidence Intervals, OLS & Ridge Regression**

4.2.7 Alternative Models: PCA-based OLS Regression

Previously, I only used those variables that appeared particularly sensible. Given potentially 2095 variables, this may not be optimal. To address the abovementioned issues, I now use PCs to capture as much variance as possible with as few regressors as possible. The implementation process and the theory behind this approach were already discussed in Sections 2.5 and 3.5. Following this approach the $MSPE_{o.o.s}$ values average approximately 65.26% of $MSPE_{Avg}$. As seen, using a PCA-based OLS Regression yields worse results than a carefully selected OLS Regression based on domain knowledge, at least when considering $MSPE_{o.o.s}$ values for model comparison, as I have done so far. However, the very important strength of this approach is that it is now possible to objectify the number of regressors used and their

makeup. Furthermore this approach still significantly outperforms Ridge Regression from a reduction of $MSPE_{o.o.s}$ per unit of additional computation time perspective. The PCA-based OLS Regression leads to a 0.7478 $MSPE_{o.o.s}$ decrease per second.

4.2.8 Alternative Models: LASSO

LASSO is the most complex alternative model to Ridge Regression that I implemented. As discussed in section 2.4.3 LASSO is a shrinkage estimator as well, which utilizes the L_1 norm instead of the L_2 norm. I implemented LASSO, only for the 2095 regressors setup, as this is the setup most similar to the setup chosen by Stock and Watson, see Stock and Watson (2020, Chapter 14, page 517). LASSO achieves $MSPE_{o.o.s}$ values averaging 39.00% of the $MSPE_{Avg}$. Therefore the results for LASSO in the 2095 regressors setup are slightly worse than the results of Ridge Regression. Furthermore LASSO comes with a notable cost, which is the necessary computation time. LASSO has the worst results, when it comes to $MSPE_{o.o.s}$ reduction per additional second of compution time. Here LASSO only lead to a 0.0092 $MSPE_{o.o.s}$ decrease per second. I even had to reduce the number of λs computed to get to reasonable computation times. Even so, the change in the number of λs has been accounted for in the $MSPE_{o.o.s}$ reduction per unit of time perspective. It is unclear what absolute $MSPE_{o.o.s}$ values could have been reached if the normal number of λs had been implemented. Still, LASSO performed very well from the absolute $MSPE_{o.o.s}$ minimization point of view and worked as a selection operator. In this specific case, LASSO set around 86.40% of the coefficients to zero, which shows that most regressors don't yield much additional explanatory power.

5 Discussion of Encountered Difficulties

Summarizing the encountered difficulties, it should be kept in mind that computation time has been a critical issue throughout this work. I had to reduce the number of λs computed for LASSO, and therefore, it is unclear what absolute $MSPE_{o.o.s}$ level would have been reached by LASSO if the normal number of λs had been computed. Comparability reduction due to a different number of λs computed is an issue. Additionally, no CIs were computed for LASSO and PCA-based OLS Regressions due to their computational demands, an issue that could be addressed in a follow-up paper. Similarily it is important to be aware of the potentially not fulfilled assumptions for the CIs, it is unclear if such a monotone transformation $m(T_n)$ exists as described in 3.6. Alternative approaches for CIs exist, like bias-corrected PBCIs or normal distribution based BCIs, but they also need assumptions where it is unclear whether they are fulfilled. Therefore, it would be worthwhile for a follow-up paper to dive further into this issue and see if a construction principle exists for which it can be shown that the necessary assumptions are fulfilled in this setup. Until this is accomplished, the CIs can only be used to get a rough idea of what differences might be structural and what differences might be due to randomness. Furthermore, all numbers concerning $MSPE_{o.o.s}$ reduction per second are based on my PC, its hardware, the code written by me, and the coding language utilized. Different hardware, an optimised version of my code or an implementation in a different programming language would lead to different results and different time constraints under which one has to operate. It should be kept in mind that many other norms could have been utilized to create and implement other shrinkage estimator and that there would have been other methods for objectifying the regressors used for an OLS based prediction, besides a PCA, like factor analysis or other sorts of optimal subset selection. Last but not least it is also important to note that the $MSPE_{o.o.s}$ or $MSPE_{i.s}$ as metric for evaluating the quality of the different models is only one option,

alternatively information criteria or measures of fit, like the R^2 could have been utilized.

6 Conclusion

In this seminar paper, I presented various methods for prediction in a cross-section data setup with many regressors. The comparison was conducted using a dataset concerning schools in California from 2013. The empirical comparison of the methods revealed that shrinkage estimators can achieve enormous $MSPE_{o.o.s}$ improvements relative to a simple OLS Regression with all variables. Even when the variables are selected based on domain knowledge, a simple OLS Regression does not reach the $MSPE_{o.o.s}$ level of the shrinkage estimators, as Figure 2 shows. It seems to be the case, that at least for this data set and these setups shrinkage estimator structurally outperform any form of OLS Regression that relies on handpicked regressors. Interestingly, a PCA-based OLS Regression also does not reach the $MSPE_{o.o.s}$ level of the shrinkage estimators, it even performs worse than some of the OLS Regressions that utilized handpicked regressors. Thus, it turns out that for prediction in a cross-section data context with many regressors, if the goal is $MSPE_{o.o.s}$ minimization, shrinkage estimators are not only a viable solution for the problems described in 2.3, but also perform structurally better than any of the other solutions discussed, at least for this data set. But if the goal is maximizing $MSPE_{o.o.s}$ reduction per unit of time and if one can allow for subjectivity and has the necessary domain knowledge utilizing an OLS Regression with handpicked regressors leads to optimal results. If subjectivity has to be eradicated for the analysis or handpicking regressors is infeasable, the next best option for maximizing $MSPE_{o.o.s}$ reduction per unit of time is a PCA-based OLS Regression.

References

Deitmer, A. (2021): *Analysis*, 3rd Edition, Springer Verlag

Friedman, J., Hastie, T. and Tibshirani R. (2010): *Regularization Paths for Generalized Linear Models via Coordinate Descent*, Journal of Statistical Software, **Volume 33**

Hansen B. E. (2022): *Econometrics*, Princeton University Press

James, G., Witten, D., Hastie, T. and Tibshirani R. (2021): *An Introduction to Statistical Learning with Applications in R*, 2nd. Edition, Springer Texts in Statistics

Johnson, R. and Wichern, D. (2014): *Applied Multivariate Statistical Analysis*, 6th. Edition, Pearson Education Limited

Kascha, C. and Trenkler, C. (2015): *Forecasting VARs, Model Selection, and Shrinkage* , Working Paper Series, **Volume 15-07**

Stock, J. H. and Watson, M. W. (2020): *Introduction to Econometrics*, 4th. Edition (Global Edition), Pearson Education Limited

Wasserman, L. (2006): *All of Nonparametric Statistics*, Springer Texts in Statistics

Wickens, T. D. (2014): *The Geometry of Multivariate Statistics*, Psychology Press