# Explaining Ridge Regression and LASSO*

Katherine Hauck† and Tiemen Woutersen‡

April 15, 2024

## 1 Introduction

Machine learning is a method that uses a computer's analytic power to make decisions and predictions from data. Two common machine learning techniques are Least Absolute Shrinkage and Selection Operator (LASSO) and Ridge regression. In some cases, these models may be preferable to least squares, and we discuss their application, implementation, and uses. We use an example to compare least squares, LASSO, and Ridge regression to demonstrate how machine learning techniques select the most important regressors for prediction analysis.

Specifically, LASSO and Ridge regression may be preferable to least squares when the researcher has a dataset with many potential explanatory variables. When the number of potential explanatory variables is much larger than the number of observations, Ridge regression or LASSO may perform well, while the least squares estimator cannot be calculated in such a case. An example of such a dataset is the type of dataset used by financial institutions to predict which potential clients are likely to make their loan payments. Such datasets include a large number of demographic variables, but it is not clear *ex ante* which of these variables are significant predictors of loan repayment. Further, some of these explanatory variables may be collinear. In such cases, machine learning techniques are able to select the subset of explanatory variables that is most important in predicting the outcome variable. In contrast, least squares uses every explanatory variable to predict the outcome variable.

The reason that researchers may be interested in using only the most important regressors in prediction analysis is related to the bias-variance trade-off. Specifically, when a researcher has many potential explanatory variables, a least squares approach

*We are grateful to Carter Hill for his enthusiasm and insights in econometrics and teaching econometrics. We wrote this paper for the volume of Advances in Econometrics in his honor. We hope that it will be helpful to teach Ridge regression and LASSO. We thank Silvia Ami Ichikawa for helpful discussions.

†University of California, Davis (email: khauck@arizona.edu)

‡University of Arizona (email: woutersen@arizona.edu)

may result in a model that is overfitted, while a machine learning technique reduces overfitting by using only the most important regressors. Therefore, machine learning techniques may outperform least squares in out-of-sample analysis. We discuss this bias-variance trade-off and its relationship to overfitting further below.

It is notable the machine learning techniques cannot distinguish correlation from causation any better than least squares can. Machine learning techniques make predictions from data and do not necessarily identify a causal relationship. A useful explanation of how machine learning techniques can fit into an economist's methodology is given by Sendhil and Spiess (2017). Taddy (2019) and Taddy, Hendrix, and Harding (2022) provide a guide to the practical implementation of machine learning and data science techniques in business, finance, and econometric settings. We also discuss how to use machine learning in causal analysis. In particular, we illustrate how to use Ridge regression and LASSO to predict counterfactuals and improving the performance of instrumental variable estimators.

# 2 Bias-Variance Trade-off

Machine learning may have a good bias-variance trade-off. Specifically, minimizing either the bias or the variance of an estimator may increase the other for the same estimator. Recall that the bias of an estimator is the difference between the expected value of the estimator and the true value of the parameter to be estimated. The variance of an estimator is the expected value of the square of the deviation from the mean, i.e., $Var(X) = \mathbb{E}[(X - \mu)^2]$, where $X$ denotes the random variable, and $\mu$ denotes the mean of $X$. As the bias of the estimator increases, the variance of the estimator may decrease, and vice versa. A researcher must find the correct balance between the size of the bias and the size of the variance of an estimator.

The bias-variance trade-off is related to overfitting and underfitting. A model that generates an estimator with a small bias and a large variance will often be overfitted, whereas a model that generates an estimator with a large bias and a small variance will often be underfitted. This relationship occurs because an estimator with a low bias means that the estimator follows the data generating process closely, but in order to do so, the estimator exhibits a large variance. Likewise, an estimator with a low variance is simple and therefore unlikely to follow the data generating process closely. An extreme example is the estimator $\hat{X} = c$, where $c$ is a constant. This estimator has $Var(c) = 0$, but it may not be an accurate estimator of a quadratic function, such as $y = x^2 + x$, i.e., this estimator would have a large bias. In such a case, the model would be underfitted.

The bias-variance trade-off illustrates a benefit of machine learning. A model that uses every explanatory variable from a possible list of 1,000 variables is likely to be overfitted. In such a case, the model would perform very well in-sample, but it would perform less well in predicting the same outcome variable out-of-sample. However, machine learning techinques such as LASSO and Ridge regression can select the most

important regressors, meaning that these models can work well out-of-sample, as well as in-sample. The Ridge regression estimator has a lower variance than the least squares estimator, but the Ridge regression estimator is biased, while the least squares estimator is unbiased.

To fix ideas, below, we use a simple example to demonstrate how a researcher balances the trade-off between the bias and the variance of an estimator. A researcher also uses this same technique in assessing machine learning models, which we discuss later in the chapter. When we covered machine learning in class, the students appreciated how simple examples can illustrate abstract concepts.

## 2.1 Dice Example to Illustrate the Bias-Variance Trade-off

Suppose a researcher is interested in estimating the average roll of a fair dice. Consider the following two estimators, $\dot{Y} = 4$ and $\ddot{Y} = \bar{Y}_{10}$, where $\dot{Y}$ denotes an estimator equal to 4, and $\ddot{Y}$ denotes an estimator equal to the average of the previous 10 rolls. To assess the bias-variance trade-off, the researcher considers the mean squared forecasting error (MSFE) of the two estimators in question. The MSFE combines the bias and variance of an estimator. Therefore, the researcher wants to minimize the MSFE. Specifically, the researcher compares

$$MSFE(\dot{Y}) = Var(Y) + Bias^2(\dot{Y}) + Var(\dot{Y}) = 2.91 + 0.25 + 0 = 3.160$$

and

$$MSFE(\ddot{Y}) = Var(Y) + Bias^2(\ddot{Y}) + Var(\ddot{Y}) = 2.91 + 0 + \frac{2.91}{10} = 3.201.$$

The above example illustrates the bias-variance trade-off. The two estimators, $\dot{Y}$ and $\ddot{Y}$, have different biases and different variances. The bias of the estimator $\dot{Y}$ is 0.5, while the bias of the estimator $\ddot{Y}$ is 0. The variance of the estimator $\dot{Y}$ is 0, while the variance of the estimator $\ddot{Y}$ is 0.291. To balance this bias-variance trade-off, the researcher must minimize the combination of the bias and the variance of each estimator using the MSFE.

The second predictor, $\ddot{Y}$, has a larger MSFE. Therefore, when the sample size (i.e., the number of previous rolls of the dice) is small, $\dot{Y}$ does better. As the sample size increases, $Var(\ddot{Y})$ decreases. Therefore, as the sample size increases, $\ddot{Y}$ becomes preferable to $\dot{Y}$.

Consider a third predictor: $\dddot{Y} = \frac{\dot{Y}+\ddot{Y}}{2}$. The variance of $\dddot{Y}$ is

$$Var(\dddot{Y}) = \frac{1}{4}\big[Var(\dot{Y}) + Var(\ddot{Y})\big] = \frac{1}{4}\big[0 + 0.29\big].$$

To calculate the bias of $\dddot{Y}$, first calculate the expected value of $\dddot{Y}$: $\mathbb{E}\Big(\frac{\dot{Y}+\ddot{Y}}{2}\Big) = \frac{4+3.5}{2} =$

3.75. Then, the bias of $\dddot{Y}$ is 3.75 - 3.5 = 0.25. Therefore, the MSFE of $\dddot{Y}$ is given by

$$MSFE(\dddot{Y}) = Var(Y) + Bias^2(\dddot{Y}) + Var(\dddot{Y}) = 2.91 + (0.25)^2 + \frac{0.29}{4} = 2.91 + 0.135 = 3.045.$$

The predictor $\dddot{Y}$ has smaller MSFE than either $\dot{Y}$ or $\ddot{Y}$, making it preferable to both of those estimators.

This simple example illustrates how averaging estimators can lower the MSFE, and we will expand on the use of the MSFE to access machine learning techniques in more complicated settings in section 4.1.

# 3 Example Comparing Least Squares, Ridge Regression, and LASSO

To demonstrate an advantage of machine learning techniques like LASSO and Ridge regression over least squares in some situations, we generate estimators to predict $Y_{X=1}$ from the same initial model, using all three techniques.

The objective of this example is to illustrate shrinkage. Shrinkage is the reduction of some parameters of a model toward zero in order to limit the effects of sampling variation. As discussed in section 2, a model may perform with high accuracy on an initial dataset but perform with less accuracy on another dataset from the same population because of overfitting. Shrinkage reduces this problem by selecting the most important regressors.

**Model**

$$Y = \alpha + \gamma X + \varepsilon$$

Let $X \in \{0, 1\}$. Assume that $\alpha$ is known. The researcher is interested in an estimator for $Y$ when $X = 1$. We generate this estimator using (i) least squares, (ii) Ridge regression, and (iii) LASSO to demonstrate the differences in each method.

**Least Squares Estimation**

$$\gamma_{LS} = \arg\min_{\gamma} \sum_i \Big(Y_i - \alpha - \gamma X_i\Big)^2$$

First order condition:

$$\sum_i X_i\Big(Y_i - \alpha - \gamma X_i\Big) = 0$$
$$\sum_{i:Xi=1} \Big(Y_i - \alpha - \gamma\Big) = 0$$
$$\gamma_{LS} = \bar{Y}_{X=1} - \alpha$$

Prediction:

$$\hat{Y}_{X=1,LS} = \bar{Y}_{X=1}$$

A benefit of any least squares estimator is that it is unbiased. However, due to the bias-variance trade-off, this estimator has a large variance when the sample size is small. Therefore, the researcher may be concerned about estimation uncertainty when using the above least squares estimator.

**Ridge Regression**

$$\gamma_{Ridge} = \arg\min_{\gamma} \sum_i \Big(Y_i - \alpha - \gamma X_i\Big)^2 + \lambda\gamma^2, \quad \lambda > 0$$

The above minimization is the same as for the least squares estimation, except that Ridge regression includes a penalty parameter $\lambda$. In the case of Ridge regression, we include this penalty term to minimize the weight of some regressors $X_i$, so that the most important regressors in predicting $Y_i$ are given the most weight. This method reduces overfitting.

For a known $\lambda$, we have the following first order condition:

$$-2\sum_i X_i\Big(Y_i - \alpha - \gamma X_i\Big) + 2\lambda\gamma = 0$$
$$\sum_{i:X_i=1} \Big(Y_i - \alpha - \gamma\Big) - \lambda\gamma = 0$$
$$N_1(\bar{Y}_{X=1} - \alpha) - (N_1 + \lambda)\gamma = 0$$
$$\gamma_{Ridge} = \frac{N_1}{N_1 + \lambda}\underbrace{\bar{Y}_{X=1} - \alpha}_{\gamma_{LS}} = \underbrace{\frac{N_1}{N_1 + \lambda}}_{<1}\gamma_{LS}$$

$N_1$ denotes the sample size. When the sample size is small, the above estimator puts more weight on $\alpha$, and when the sample size is large, the estimator puts more weight on $\hat{Y}_{X=1,LS} = \bar{Y}_{X=1}$. Therefore, when the sample size is large, the Ridge regression estimator is close to the (unbiased) least squares estimator. Least squares

performs best when the sample size is large, so a benefit of Ridge regression is that it approximates least squares when the sample size is large. Meanwhile, when the sample size is small, the Ridge regression estimator uses relatively more data from $X = 0$ (in addition to data from $X = 1$) in predicting $\bar{Y}_{X=1}$ to make up for the small sample size.

When using least squares, $\hat{Y}_{X=1,LS}$ is solely based on the subset of the data where $X = 1$. However, when using Ridge regression, $\gamma_{Ridge}$ is a weighted average of 1) the data when $X = 1$ and 2) all the data. Further, using Ridge regression, when $N_1$ is small, more weight is given to the overall average from the full dataset $\bar{Y}$. When $N_1$ is large, more weight is given to the specific average $\bar{Y}_{X=1}$.

An important advantage of Ridge regression over least squares (or LASSO, discussed below), is that Ridge regression can handle perfect collinearity in the potential regressors, while least squares cannot. Specifically, when $Y_i = \gamma W_i + \delta W_i + \varepsilon_i$, least squares does not have a solution. In such a case, multiple combinations of $\gamma$ and $\delta$ give the same solution:

$$\sum_i \Big(Y_i - (\gamma + \kappa)W_i - (\delta - \kappa)W_i\Big)^2 = \sum_i \Big(Y_i - \gamma W_i - \delta W_i\Big)^2$$

Least squares relies on the assumption of no perfect collinearity. When this assumption does not hold, as above, $\gamma_{LS}$ and $\delta_{LS}$ do not exist. There exist many solutions to the minimization problem. However, Ridge regression can handle such a case, and it provides stable and feasible estimates: $\gamma_{Ridge} = \delta_{Ridge}$.

**LASSO**

$$\gamma_{LASSO} = \arg\min_{\gamma} \sum_i \Big(Y_i - \alpha - \gamma X_i\Big)^2 + \lambda|\gamma|, \quad \lambda > 0$$

Like Ridge regression, LASSO includes a penalty parameter $\lambda$. LASSO balances the number of regressors $X_i$ with the power of those regressors to predict $Y_i$ by dreopping the least important regressors. This method reduces overfitting.

First, consider the case where $|\gamma| = \gamma$. In such a case, $\gamma_{LS} = \bar{Y}_{X=1} - \alpha$ is large.

For a known $\lambda$, we have the following first order condition:

$$\begin{aligned}
-2\sum_i X_i\Big(Y_i - \alpha - \gamma X_i\Big) + \lambda &= 0 \\
\sum_{i:X_i=1}\Big(Y_i - \alpha - \gamma\Big) - \frac{\lambda}{2} &= 0 \\
N_1(\bar{Y}_{X=1} - \alpha) - \frac{\lambda}{2} - N_1\gamma &= 0 \\
\gamma_{LASSO} = \bar{Y}_{X=1} - \alpha - \frac{\lambda}{2N_1}
\end{aligned}$$

Now, consider the case where $|\gamma| = -\gamma$. This case results in:

$$\gamma_{LASSO} = \bar{Y}_{X=1} - \alpha + \frac{\lambda}{2N_1}$$

Therefore, LASSO results in the following estimator:

$$\gamma_{LASSO} = \begin{cases} \bar{Y}_{X=1} - \alpha - \frac{\lambda}{2N_1}, & \bar{Y}_{X=1} - \alpha \geq \frac{\lambda}{2N_1} \\ 0, & -\frac{\lambda}{2N_1} < \bar{Y}_{X=1} - \alpha < \frac{\lambda}{2N_1} \\ \bar{Y}_{X=1} - \alpha + \frac{\lambda}{2N_1}, & \bar{Y}_{X=1} - \alpha < -\frac{\lambda}{2N_1} \end{cases}$$

LASSO and Ridge regression both perform better than least squares when either (i) the number of potential regressors is greater than the number of observations, or (ii) the variance of the least squares estimator is large.

While both Ridge regression and LASSO can out-perform least squares in some situations, Ridge regression may fit some situations better than LASSO and vice versa. In particular, Ridge regression is preferable to LASSO when the researcher has a dataset with perfect multicollinearity (or high collinearity) in the potential explanatory variables. Further, Ridge regression can handle including some regressors twice. On the other hand, the LASSO estimation results may be easier to explain. In particular, LASSO selects only some potential explanatory variables and drops the others, while Ridge regression shrinks all regressors toward zero but does not drop any. Both methods reduce overfitting, meaning they can perform well out-of-sample, but the LASSO results may be easier to explain to a non-technical audience because the number of coefficients will be smaller.

## 3.1 Motivating Example: Gift Card

In this section, we lay out an empirical example in which a researcher has a question that can be answered using Ridge regression or LASSO. Consider a situation in which a company randomly gives some survey participants a 20 dollar gift card. Suppose a

researcher is interested in the counterfactual situation in which the individuals who received the gift card did not recieve it.

Formally, let $X$ denote the treatment: $X_i = 1$ if individual $i$ gets a 20 dollar gift card, and $X_i = 0$ if they don't. In the data, there are 2,000 individuals with $X = 1$ and millions of individuals with $X = 0$. Let $Y$ denote the outcome of interest to the researcher. Let the vector $W$ denote the other regressors, and let $K$ denote the number of regressors, where $K = 100,000$. Because $K$ is large, Ridge regression or LASSO is preferable to least squares.

Suppose the researcher is interested in the following counterfactual: what would have happened to the outcome $Y$ for the $X = 1$ group if the $X = 1$ group did not receive a gift card?

To evaluate this counterfactual, use the sample with $X = 0$ to estimate $\beta_{Ridge}$ from the model $Y = W\beta + \varepsilon$. The counterfactual is $\hat{Y}_{i,X=0} = W_i\beta_{Ridge}$. The researcher can use this counterfactual to estimate the effect of the gift card with

$$\frac{1}{2000} \sum_{i:X_i=1} (Y_i - \hat{Y}_{i,X=0}).$$

In other words, the effect of the gift card is calculated by the average difference in the observed outcome (with the gift card) and the predicted outcome (without the gift card) for only $X = 1$ individuals. Ridge regression is used to estimate the predicted outcome without the gift card.

# 4 Training, Validation, and Testing Samples

To implement machine learning techniques such as Ridge regression and LASSO, the researcher splits the data into three subsamples. These subsamples are used for training, validation, and testing. A common split of the data is to use 60% of the data in the training sample and 20% of the data each in the validation and testing samples. The reason for splitting the data into subsamples is to assess the fit of the model. Splitting the data reduces overfitting for better out-of-sample predictions.

First, the training sample is used to initially estimate the parameters of the model, based solely on the data in this subsample. In this step, the tuning parameter $\lambda$ is taken as given. Second, the model that was estimated on the training subsample is applied to the validation subsample. In this step, the researcher uses the model to predict the data in the validation subsample, and then compares these predictions to the actual data in the validation subsample. This step provides the researcher with information about how accurately the model predicts the data and can also indicate if the model is overfitted. If the model performs well in the training dataset and poorly in the validation dataset, that may indicate that the model is overfitted. Finally, the model is estimated on the testing subsample to generate final estimates from the model.

A common question is why the researcher needs to split the data into both training and validation samples. Note that least squares estimation minimizes the sum of square residuals, which means that — by construction — least squares is an excellent in-sample predictor. Therefore, in order to assess the model's predictive power out-of-sample, the researcher must use both training and validation samples. Specifically, the training sample is used to create several different models, the validation sample is used to choose the best model among those, and the testing sample is used to generate the final MSFE estimates.

Below, we demonstrate how using this process can select the best penalty parameter $\lambda$ in a Ridge regression model.

## 4.1 Ridge Regression: Choosing the Penalty Parameter $\lambda$

One common use of training, validation, and testing samples in machine learning is to choose the penalty parameter $\lambda$ in Ridge regression. The choice of $\lambda$ determines the Ridge regression estimator $\gamma_{Ridge}(\lambda)$. Because of the bias-variance trade-off, the researcher uses the training, validation, and testing samples to select the $\lambda$ that minimizes the MSFE for $\gamma_{Ridge}(\lambda)$. The following example illustrates how to choose $\lambda$ using this method.

**Model**

$$Y = \alpha + \gamma W + \epsilon$$

Let $W \in \{0, 1\}$, $\lambda > 0$, and assume $\alpha$ is known. Suppose the researcher is interested in estimating $\bar{Y}_{X=1}$. Following section 3, the Ridge estimator, as a function of $\lambda$, is given by

$$\gamma_{Ridge}(\lambda) = (\bar{Y}_{W=1} - \alpha)\frac{N}{N + \lambda},$$

where $N$ is the sample size. Further, as in section 3, the least squares estimator is $\gamma_{LS} = \bar{Y}_{W=1} - \alpha$. Then, as in section 3, we can rewrite the Ridge estimator as

$$\gamma_{Ridge}(\lambda) = \underbrace{\bar{Y}_{W=1} - \alpha}_{\gamma_{LS}} - (\bar{Y}_{W=1} - \alpha)\frac{\lambda}{N + \lambda}.$$

**Bias-Variance Trade-off**

Recall that the Ridge regression estimator has lower variance than the least squares estimator, but the Ridge regression estimator is biased, while the least squares estimator is unbiased. The bias and variance of the Ridge regression estimator are given

by

$$Var\{\gamma_{Ridge}(\lambda)\} = \left(\frac{N}{N+\lambda}\right)^2 Var(\gamma_{LS}) < Var(\gamma_{LS})$$

and

$$Bias\{\gamma_{Ridge}(\lambda)\} = \gamma\frac{\lambda}{N+\lambda} > 0.$$

Because of the bias-variance trade-off, a choice of $\lambda$ that minimizes the bias of the Ridge regression estimator $\gamma_{Ridge}(\lambda)$ will also increase the variance of the same estimator, and vice versa. Therefore, the researcher wants to choose $\lambda$ in order to generate the $\gamma_{Ridge}(\lambda)$ that has the lowest MSFE. We describe how to choose this $\lambda$ below.

**Choosing $\lambda$**

The researcher wants to choose the penalty parameter $\lambda$ in such a way that it achieves a good bias-variance trade-off. In other words, the researcher chooses $\lambda$ according to minimize the MSFE of $\gamma_{Ridge}(\lambda)$, just as in the dice example in Section 2.1.

In order to use machine learning techniques to choose $\lambda$ to minimize the MSFE, the researcher first splits the data in three groups: training sample (60% of the data), validation sample (20% of the data), and testing sample (20% of the data). Then, the researcher uses the following steps:

1. Using the training sample: Estimate $\gamma_{Ridge}(\lambda)$ for different values of $\lambda$. For example, estimate $\gamma_{Ridge}(\lambda)$ for $\lambda = 0.5$, $\lambda = 1$, etc.

2. Using the validation sample: Calculate the MSFE using $\gamma_{Ridge}(\lambda = 0.5)$, $\gamma_{Ridge}(\lambda = 1)$, ... and pick the one with the smallest MSFE.

3. Using the testing sample: Calculate the MSFE using the best $\gamma_{Ridge}(\lambda^*)$ (the one with the smallest MSFE) from step 2.

The above example demonstrates why the researcher needs to split the data into both training and validation samples. Specifically, because least squares estimation minimizes the sum of square residuals, it results in the best in-sample predictor. Therefore, setting $\lambda = 0$ (i.e, no shrinkage, just using least squares), results in the lowest MSFE. However, the researcher is interested in out-of-sample prediction. Therefore, it is necessary to try multiple values for $\lambda$ and assess the resulting MSFE in multiple samples.

**Rotating the Split of the Data**

Above, we provided a simple algorithm to use machine learning techniques to pick the best penalty parameter $\lambda$ in a Ridge regression model. This $\lambda$ minimizes the

MSFE of the Ridge regression estimator. The following procedure makes a slight improvement to this algorithm. Instead of splitting the data once, the researcher rotates the split of the training and the validation samples four times. This method improves on the above algorithm because it provides a more robust and stable result due to using multiple combinations of the split of the data.

First, the researcher splits the data in two groups: 80% of the data will rotate between the training and validation samples, and 20% of the data remains as the testing sample. For example, using a dataset with $N = 100,000$ observations, $i = 1, ..., 80,000$ are used for training and validation, and $i = 80,001, ..., 100,000$ are used for testing. Then, the researcher repeats this split three more times. The following algorithm illustrates an example of this technique.

1. Take 80% of the data (training sample + validation sample), and split it four times:

   (I) First 20% $\rightarrow$ Validation sample. In the example, we use $i = 1, ..., 20,000$ as the validation sample and $i = 20,001, ..., 80,000$ as the training sample.

   (II) Second 20% $\rightarrow$ validation sample. We use $i = 20,001, ..., 40,000$ as the validation sample and $i = 1, ..., 20,000 + i = 40,001, ..., 100,000$ as the training sample.

   (III) Third 20% $\rightarrow$ validation sample. We use $i = 40,001, ..., 60,000$ as the validation sample and $i = 1, ..., 40,000 + i = 60,001, ..., 100,000$ as the training sample.

   (IV) Fourth 20% $\rightarrow$ validation sample. We use $i = 60,001, ..., 80,000$ as the validation sample and $i = 1, ..., 60,000 + i = 80,001, ..., 100,000$ as the training sample.

2. Follow these steps to select the penalty parameter $\lambda$:

   1. Using the training sample: Estimate $\gamma_{Ridge}(\lambda)$ for each value of $\lambda$ four separate times, i.e., estimate $\gamma_{Ridge}(\lambda)$ using the training samples for I, II, III, IV. For example, estimate $\gamma_{Ridge}(\lambda)$ for $\lambda = 0.5$, $\lambda = 1$, etc. in each of the training samples I, II, III, IV.

   2. Using the validation sample: Calculate the the MSFE four times using $\gamma_{Ridge}(\lambda)$, and use the *average* MSFE to choose the best value of $\lambda$. For example, calculate the MSFE across all four splits for $\gamma_{Ridge}(\lambda = 0.5)$ in each of the four samples I, II, III, and IV, and average over all four samples to find the average MSFE for $\gamma_{Ridge}(\lambda = 0.5)$. Compare this average MSFE to the average MSFE for $\gamma_{Ridge}(\lambda = 1)$, etc.

   3. Using the testing sample: Calculate MSFE using the best $\gamma_{Ridge}(\lambda^*)$ (the one with the smallest average MSFE) from step 2.

# 5 Ridge Regression Using Matrix Notation

A useful feature of Ridge regression is that it generates an estimate that is an explicit function of the data. We now demonstrate this feature in a situation in which the researcher is interested in an outcome vector instead of a single outcome variable. This situation is a generalization of the Ridge regression example in Section 3. As in Section 3, we estimate the same initial model using both (i) least squares and (ii) Ridge regression to demonstrate the differences between the methods. Specifically, least squares cannot handle perfect collinearity, but the Ridge estimator can.

Consider the following model in matrix notation,

$$Y = X\beta + \varepsilon,$$

where $Y$ is $(N \times 1)$, $X$ is $(N \times K)$ and $\beta$ is $(K \times 1)$. Specifically, the definitions of $X$, $\beta$, and $Y$ are given below.

$$X = \begin{pmatrix} 1 & S_1 \\ 1 & S_2 \\ \vdots & \vdots \\ 1 & S_N \end{pmatrix} \qquad \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \qquad Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix}$$

The researcher observes $\{X_i, Y_i\}_{i=1}^N$. The residual is $e = Y - X\beta$.

**Least Squares Estimation**

$$\begin{aligned} \beta_{LS} &= \arg\min_{\beta} (Y - X\beta)'(Y - X\beta) \\ &= \arg\min_{\beta} Y'Y - \beta'X'Y - Y'X\beta + \beta'X'X\beta \\ &= \arg\min_{\beta} Y'Y - 2\beta'X'Y + \beta'X'X\beta \end{aligned}$$

First order condition (with respect to $\beta'$):

$$-2X'Y + 2X'X\beta = 0$$

This yields the least squares estimator:

$$\beta_{LS} = (X'X)^{-1}X'Y$$

**Ridge Regression**

$$\begin{aligned}
\beta_{Ridge} &= \arg\min_{\beta}(Y - X\beta)'(Y - X\beta) + \lambda \underbrace{\beta'\beta}_{\sum_k \beta_k^2} \quad \lambda > 0 \\
&= \arg\min_{\beta} Y'Y - \beta'X'Y - Y'X\beta + \beta'X'X\beta + \lambda\beta'\beta \\
&= \arg\min_{\beta} Y'Y - 2\beta'X'Y + \beta'X'X\beta + \lambda\beta'\beta \\
&= \arg\min_{\beta} Y'Y - 2\beta'X'Y + \beta'(X'X + \lambda I)\beta
\end{aligned}$$

First order condition (with respect to $\beta'$):

$$\begin{aligned}
-2X'Y + 2(X'X + \lambda I)\beta &= 0 \\
-2(X'X + \lambda I)^{-1}X'Y + 2(X'X + \lambda I)^{-1}(X'X + \lambda I)\beta &= 0
\end{aligned}$$

This yields the Ridge regression estimator

$$\beta_{Ridge} = (X'X + \lambda I)^{-1}X'Y,$$

where $\lambda > 0$. In this case, $\beta_{LS}$ and $\beta_{Ridge}$ are both $(K \times 1)$ vectors of parameters. Given the definitions of $X$, $\beta$, and $Y$ above, $X'Y$ is equal to

$$X'Y = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ S_1 & S_2 & \cdots & S_N \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} \sum_i Y_i \\ \sum_i S_i Y_i \end{pmatrix}.$$

Further, differentiation of $X'Y$ with respect to a vector $\beta'$ gives

$$\begin{aligned}
\beta'X'Y &= \left(\beta_1 \textstyle\sum_i Y_i + \beta_2 \sum_i S_i Y_i\right) \\
\frac{\partial \beta'X'X}{\partial \beta_1} &= \sum_i Y_i, \qquad \frac{\partial \beta'X'X}{\partial \beta_2} = \sum_i S_i Y_i.
\end{aligned}$$

To illustrate a benefit of Ridge regression over least squares, consider a case in which some of the potential regressors are perfectly collinear. Recall that Ridge regression can still be used in a situation with perfect collinearity, while least squares and LASSO cannot.

We first consider the case where $S_i = 1$ for all $i$. An example of such a case is universal treatment without a control group. Having only a treatment group means that one cannot compare the treatment group to a control group. In such a case, there is no value of the parameter that uniquely minimizes the sum of squared residuals; therefore, the least squares estimator does not exist. Similarly, the LASSO

estimator does not exist. Intuitively, however, one can still predict the outcome for the treatment group, and the Ridge regression estimator yields such a prediction.

To demonstrate how Ridge regression and least squares operate in a situation in which $S_i = 1$ for all $i$, recall that the least squares estimator is given by

$$\beta_{LS} = (X'X)^{-1}X'Y.$$

When $S_i = 1$ for all $i$, $X'X$ is a matrix in which all the entries are 1, and $(X'X)^{-1}$ does not exist. Therefore, $\beta_{LS}$ does not exist.

Compare this problem with least squares when $S_i = 1$ for all $i$ to the Ridge estimator when $S_i = 1$ for all $i$. Recall that the Ridge regression estimator is given by

$$\beta_{Ridge} = (X'X + \lambda I)^{-1}X'Y,$$

where $\lambda > 0$. Because of the $\lambda I$ term, $\beta_{Ridge}$ is identified. Specifically, $\lambda I = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$ Therefore, when $S_i = 1$ for all $i$, $X'X + \lambda I$ is given by

$$\begin{pmatrix} \lambda + 1 & 1 \\ 1 & \lambda + 1 \end{pmatrix}.$$

This matrix does have a unique inverse, which means that $\beta_{Ridge}$ is identified.

Now consider the more general case of perfect collinearity, in which $X = \begin{pmatrix} S_1 & W_1 \\ \vdots & \\ S_N & W_N \end{pmatrix}$ and $S = W$. To demonstrate how Ridge regression and least squares operate in this more general version of perfect collinearity, we begin by generating the least squares estimator for this case. Specifically, recall that the least squares estimator is

$$\beta_{LS} = (X'X)^{-1}X'Y.$$

When $X = \begin{pmatrix} S_1 & W_1 \\ \vdots & \\ S_N & W_N \end{pmatrix}$ and $S = W$, then $X'X$ is

$$X'X = \begin{pmatrix} \sum_i S_i^2 & \sum_i S_i W_i \\ \sum_i S_i W_i & \sum_i W_i^2 \end{pmatrix} = \begin{pmatrix} \sum_i S_i^2 & \sum_i S_i^2 \\ \sum_i S_i^2 & \sum_i S_i^2 \end{pmatrix}.$$

When $S = W$, $X'X$ does not have a unique inverse, so the identification of $\beta_{LS}$ fails.

Conversely, the Ridge estimator is still identified when $S = W$. We show this result below. Recall that the Ridge estimator is given by

$$\beta_{Ridge} = (X'X + \lambda I)^{-1}X'Y,$$

where $\lambda > 0$. The $\lambda I$ term is necessary for identification of $\beta_{Ridge}$. As in the case of $S_i = 1$ for all $i$, $\lambda I = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$, which means that the inverse of $(X'X + \lambda I)$ is

$$\left( \begin{pmatrix} \sum_i S_i^2 & \sum_i S_i^2 \\ \sum_i S_i^2 & \sum_i S_i^2 \end{pmatrix} + \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right)^{-1} = \frac{1}{(\sum_i S_i^2 + \lambda)^2 - (\sum_i S_i)^2} \cdot \begin{pmatrix} \sum S_i^2 + \lambda & -\sum S_i^2 \\ -\sum S_i^2 & \sum S_i^2 + \lambda \end{pmatrix}.$$

This matrix has a unique inverse, so $\beta_{Ridge}$ is identified.

These two examples illustrate a benefit of Ridge regression over least squares: Ridge regression can handle perfect collinearity in the potential regressions, while least squares cannot.

Earlier, we discussed how Ridge regression and LASSO can be used to predict counterfactuals using the gift card example. Another important tool to deal with endogeneity is instrumental variables. When using instrumental variables in the linear model, a researcher replaces the endogenous variables by predictions that are constructed using instruments. Using least squares is the most popular way to generate these predictions. Hausman, Newey, Woutersen, Chao, and Swanson (2012) propose to use Ridge regression for this prediction instead of least squares. This procedure allows for perfect multicollinearity of the instruments, just as Ridge regression allows for perfect multicollinearity in other prediction contexts, as we discussed above.

# 6 Conclusion

This chapter discusses the uses, application, and implementation of machine learning techniques like LASSO and Ridge regression and their benefits over least squares in some situations. A major use of LASSO and Ridge regression is in cases in which the researcher has a dataset with many more potential explanatory variables than observations and wants to select the most important explanatory variables, i.e., the regressors with the most power in predicting the outcome variable $Y$. Another important application of LASSO and Ridge regression is when the number of regressors is large but smaller than the sample size. In that case, the least squares estimator would likely overfit the data, and it would not perform well in out-of-sample prediction. In constrast, a technique such as LASSO or Ridge regression will result in a somewhat biased estimator, but the estimator uses only the regressors that are most powerful in predicting the outcome variable. Therefore, such an estimator performs better in out-of-sample prediction than least squares.

A researcher's decision to use a technique such as LASSO or Ridge regression is related to the bias-variance trade-off of an estimator. This trade-off refers to the fact that minimizing either the bias or the variance of an estimator increases the other for the same estimator. Because of this trade-off, a researcher minimizes the MSFE by balancing the bias and the variance of an estimator. Specifically, a least

squares estimator will be unbiased, but it may have a high variance. LASSO or Ridge regression will result in a biased estimator, but the variance may be lower than the variance of the least squares estimator, meaning that the overall MSFE of the LASSO or Ridge regression estimator may be lower.

We use an example to compare least squares, Ridge regression, and LASSO. Ridge regression is preferable to LASSO when the potential explanatory variables have perfect multicollinearity, whereas LASSO is preferable to Ridge regression when the researcher wants to drop some potential explanatory variables completely.

To assess the fit of the model, machine learning techniques split the data into three subsamples: training, validation, and testing. A common split of the data is to use 60% of the data in the training sample and 20% of the data each in the validation sample and the testing sample. A useful application of this technique is to choose the best penalty parameter $\lambda$ in a Ridge regression or LASSO model. The ideal penalty term is selected to minimize the MSFE.

We also discuss how to use machine learning in causal analysis. In particular, we discuss how using Ridge regression and LASSO predicts counterfactuals and improves the performance of instrumental variable estimators.

# 7 Sample Questions

1. Let $Y \in \{0, 1\}$ denote the flip of a fair coin. What is the MSFE of a predictor $\hat{Y} = \bar{Y}$? What is the MSFE of the predictor $\hat{Y} = 0.4$?

2. Follow the example in Section 2.1, does the predictor $\frac{\dot{Y}+\ddot{Y}}{2}$ have smaller MSFE than $\dot{Y}$?

3. Consider the model
$$Y_i = \alpha + \gamma X_i + \varepsilon_i$$
Assume $\alpha$ is known. Predict $\hat{Y}_{X=1}$ using $\gamma_{LS}, \gamma_{LASSO}$, and $\gamma_{Ridge}$.

4. When do you prefer $\gamma_{LASSO}$ versus $\gamma_{LS}$? $\gamma_{Ridge}$ versus $\gamma_{LS}$? $\gamma_{Ridge}$ versus $\gamma_{LASSO}$?

5. Let
$$Y_i = \alpha + \gamma W_i + \varepsilon_i,$$
where $W_i = 1$ for all $i$. Let $\bar{Y} = 1$. What are your estimates for $\alpha_{Ridge}$ and $\gamma_{Ridge}$ if $\lambda = 1$?

6. Let
$$Y_i = \alpha + \gamma_1 W_{1i} + \gamma_2 W_{2,i} + ... + \gamma_{1000} W_{1000,i} + \varepsilon_i,$$
where $N = 3000$.

   (a) Explain how you would use LASSO to estimate the parameters.

(b) What would happen if you used 80% of your sample for training, then use this 80% again for validation, and finally, use the remaining 20% to calculate the MSFE?

(c) What would happen if you used 60% of your sample for training, then the use 40% of your sample for validation and use that same 40% to calculate the MSFE?

# 8 References

Hausman, J. A., W. K. Newey, T. Woutersen, J. C. Chao, and N. R. Swanson (2012): "IV Estimation with Heteroscedasticity and Many Instruments," *Quantitative Economics*, 3, 211–255.

Sendhil Mullainathan and J. Spiess (2017): "Machine Learning: An Applied Econometric Approach." *Journal of Economic Perspectives*, 87-106.

Taddy, Matt (2019): *Business Data Science: Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions,* 1st Edition, McGraw-Hill, New York, NY.

Taddy, Matt, Leslie Hendrix, and Matthew Harding (2022): *Modern Business Analytics*, McGraw-Hill, New York, NY.