Asian Journal of Economics and Banking
ISSN 2615-9821
http://ajeb.buh.edu.vn/Home

How to Choose Tuning Parameters in Lasso and Ridge Regression?

Chon Van Le†

International University, Vietnam National University, Ho Chi Minh City, Quarter 6, Linh Trung, Thu Duc Dist. Ho Chi Minh City, Vietnam.

Article Info

Received: 10/02/2020
Accepted: 16/3/2020
Available online: In Press

Keywords
Tuning parameter, Cross-validation, Bootstrap, AIC, BIC.

JEL classification
C10, C61, R31

MSC2010 classification
62H12, 62J07, 62M20

Abstract

This paper gives a literature review of choosing tuning parameters for ridge regression and lasso. These regularized regressions introduce a little bias in return for a considerable decrease in the variance of the predicted values, thus increasing prediction accuracy. We can use AIC or BIC to select the tuning parameter for linear predictive models. However, for general predictive models, cross-validation and bootstrap work better because they directly estimate prediction error. Though, cross-validation is more widely used than bootstrap. An empirical example is employed to illustrate how cross-validated lasso works. It shows that lasso may solve the multicollinearity problem.

†Corresponding author: Chon Van Le. International University, Vietnam National University, Ho Chi Minh City, Quarter 6, Linh Trung, Thu Duc Dist. Ho Chi Minh City, Vietnam. Email: lvchon@hcmiu.edu.vn

1 INTRODUCTION

The method of least squares, which was invented in early 1800's (see [18]), has been the power horse of modern statistical analysis. Its merit includes the intuitive rationale of "best fit" of the sample regression function to a given sample data set and the closed-form estimator that have several desirable statistical properties. When the classical linear regression model (CLRM) assumptions are satisfied, the least squares estimator is the minimum variance linear unbiased estimator of the population parameter vector. It is also consistent in large, well-behaved data sets and is asymptotically normal as a consequence of the central limit theorem (see [10]). Its asymptotic normality allows hypothesis testing and interval estimation in statistical inference. The least squares method has received various extensions which are applied when the CLRM assumptions are untenable.

Since the least squares coefficient estimates are never zero, independent variable selection is based solely on the classical Neyman-Pearson null hypothesis testing procedure. This methodology has, however, become invalid following the conclusion of the American Statistical Association (ASA) in March 2019 that a declaration of "statistical (in)significance" is now meaningless (see [22]). If the number of explanatory variables is greater than the sample size, the least squares fails to produce a unique solution. In addition, the least squares estimate often has low bias but large variance due to multicollinearity, which can deteriorate prediction accuracy as measured in terms of the mean squared error (see [13]).

By shrinking the values of the regression coefficients, regularized versions of the least squares introduce a little bias but might lead to a substantial decrease in the variance of the predicted values, hence improving the overall prediction accuracy. A least squares regression model subject to an L_2-norm constraint of the parameter vector is called ridge regression, whereas a model subject to an L_1-norm constraint is called lasso (least absolute shrinkage and selection operator). One of the key differences between ridge regression and lasso is that in ridge regression, as the constraint gets tighter, all coefficients are reduced but remain non-zero, while in lasso, imposing a tighter constraint will cause some coefficients equal to zero. This is an advantage of lasso in variable selection as the classical hypothesis testing procedure no longer works.

An issue for regularized regression is how much constraint should be placed on the coefficient vector. The amount of regularization is controlled by the tuning parameter, so it is crucial to choose a good value of the tuning parameter. Because each tuning parameter is associated with a different fitted model, even with a different set of independent variables in lasso, the choice of its value is also referred to as model selection. It however depends on our purposes of prediction or causality analysis. This paper gives a literature review of choosing the tuning parameter for the former purpose. Some information criteria such as Akaike Information Criteria

(AIC) and Bayesian Information Criteria (BIC), and Mallows's C_p statistic, which are used to assess the fit of a regression model, can work because their training error is already adjusted to estimate prediction error. But their usage is restricted to estimates that are linear in their parameters. Cross-validation and bootstrap methods, which are direct estimates of extra-sample error, are considered better ways to choose the tuning parameter.

The paper is structured as follows. Section 2 introduces ridge regression and lasso. Section 3 discusses AIC, BIC, and Mallows's C_p statistic as selection criteria. Sections 4 and 5 present cross-validation and bootstrap methods, respectively. Section 6 gives an empirical example using cross-validated lasso with Stata commands explained in the Appendix. Conclusions follow in Section 7.

2 RIDGE REGRESSION AND LASSO

In an unconstrained least squares estimation, multicollinearity can cause coefficient estimates to explode and hence susceptible to very high variance. This problem is mitigated when a size constraint is imposed on the coefficients. Ridge regression (see [14]) uses the same least squares objective function, but adds an L_2-norm constraint on the magnitude of regression coefficients. The minimization problem of penalized residual sum of squares is

$$\min_{\boldsymbol{\beta}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \quad (1)$$
$$\text{subject to } \boldsymbol{\beta}^T\boldsymbol{\beta} \leq t,$$

where t is the size constraint on the parameters. The ridge estimator can be equivalently written as

$$\hat{\boldsymbol{\beta}}_{ridge} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\big[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}\big]. \quad (2)$$

Lagrangian duality implies that there is a one-to-one correspondence between the constrained optimization (1) and the Lagrangian form (2). Specifically, there is a corresponding value of λ for each value of t such that both (1) and (2) yield the same solution. The solution is

$$\hat{\boldsymbol{\beta}}_{ridge} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}, \quad (3)$$

where $\boldsymbol{I}$ is the identity matrix, the Lagrange multiplier λ is nonnegative and called a tuning or shrinkage parameter because it controls the amount of shrinkage. A larger λ indicates a greater amount of shrinkage. When $\lambda = 0$, there is no shrinkage and we obtain the least squares solution. As λ increases, the coefficients are shrunk toward zero.

Since measurement scale of covariates $\boldsymbol{X}$ affects ridge estimates, we normally standardize independent variables before solving (1). In so doing, the intercept β_0 is not included in the shrinkage. The solution (3) is again a linear function of $\boldsymbol{y}$. It adds a positive constant λ to the diagonal of $\boldsymbol{X}^T\boldsymbol{X}$ before inversion. When the number of parameters is greater than the sample size, $\boldsymbol{X}^T\boldsymbol{X}$ is singular and the least squares solution is not unique. On the contrary, $(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})$ is always of full rank, hence the ridge solution is unique. Hoerl and Kennard [14] considered it the original motivation for ridge regression. In addition, when the number

of parameters equals the sample size, least squares regression fits data "too well", i.e., there is lack of generalization. Ridge regression does not overfit data with a proper value of λ. However, the main drawback of ridge regression is that its estimated coefficients, although being shrunk, are never zero. It makes subset selection difficult especially when the classical hypothesis testing procedure is not valid. It is nicely settled by using a different penalty on the coefficients in an alternative version of shrinkage regression named lasso.

Least absolute shrinkage and selection operator, or lasso for short, was developed by Tibshirani [20]. Lasso forces the sum of the absolute values of the regression coefficients to be less than a fixed value t. Its objective function is

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \quad (4)$$
$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t.$$

The lasso problem can be rewritten in the Lagrangian form

$$\hat{\boldsymbol{\beta}}_{lasso} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \quad (5)$$

Like in ridge regression, explanatory variables are standardized, thus excluding the constant β_0 from (5).

Lasso differs from ridge regression in that it uses an L_1-norm instead of an L_2-norm. The L_1 penalty makes the lasso solution nonlinear in $\boldsymbol{y}$, and there is no closed form expression as in ridge regression. Since the constraint region is still convex, lasso possesses all strengths of ridge regression. The lasso solution is unique and lasso provides a better fit in high dimensional data where the number of features can exceed the number of observations. Because the constraint region is diamond-shaped, a sufficiently large value of λ is likely to pick a solution that lies at a corner point of that region. As a result, we have a sparse solution in which some coefficients are set exactly equal to zero. In other words, lasso performs a straightforward model selection. This distinct advantage of lasso over ridge regression explains its popularity.

Nevertheless, it should be noted that when lasso is used in building models for prediction, it does not necessarily choose the independent variables that belong in the true model, but it chooses a set of variables that are correlated with them. That a potential variable is omitted does not tell whether it belongs in the true model or not but implies that it is correlated with variables that are already selected. Those variables are included because they are useful for prediction which is our interest. The main parameter that a researcher has to choose is the tuning parameter λ.

3 MALLOWS'S C_P STATISTIC, AIC, AND BIC

To understand why Mallows's C_p statistic, AIC, and BIC can be used

[a] This part is based on Hastie et al. [12].

as selection criteria for λ, let us examine the optimism of the training error rate[a]. Suppose we have a response y, and a vector of regressors $\boldsymbol{x}$, and a prediction model $\hat{f}(\boldsymbol{x})$ that is estimated from a training set $\mathcal{T} = \{(y_1, \boldsymbol{x}_1), (y_2, \boldsymbol{x}_2), \ldots, (y_N, \boldsymbol{x}_N)\}$. The loss function that measures squared errors between y and $\hat{f}(\boldsymbol{x})$ is

$$L(y, \hat{f}(\boldsymbol{x})) = (y - \hat{f}(\boldsymbol{x}))^2. \quad (6)$$

Generalization error, also referred to as test error, of the model $\hat{f}$ given the fixed training set $\mathcal{T}$ is

$$\mathrm{Err}_{\mathcal{T}} = E_{y', \boldsymbol{x}'}[L(y', \hat{f}(\boldsymbol{x}'))|\mathcal{T}], \quad (7)$$

where $(y', \boldsymbol{x}')$ is a new test data point that is drawn from the joint distribution of the response and regressors. The expected test error (or expected prediction error) is averaged over training sets $\mathcal{T}$

$$\mathrm{Err} = E_{\mathcal{T}} E_{y', \boldsymbol{x}'}[L(y', \hat{f}(\boldsymbol{x}'))|\mathcal{T}]. \quad (8)$$

Our goal is to estimate conditional error $\mathrm{Err}_{\mathcal{T}}$, but in most cases it is easier to estimate the expected error Err.

Training error is the average loss in the training set

$$\overline{\mathrm{err}} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}(\boldsymbol{x}_i)). \quad (9)$$

Training error is not a good estimate of the test error $\mathrm{Err}_{\mathcal{T}}$. When the model gets more and more complex, it is likely to extract some of the residual variation which should be considered noise as if that variation represents certain underlying structures. In other words, the model begins to "memorize" training data rather than "learning" to generalize from a trend. Therefore, the model can predict the training data perfectly, but typically fails severely on unseen data. Overfitting would cause the training error to decrease consistently and to be less than the true test error $\mathrm{Err}_{\mathcal{T}}$. Or $\overline{\mathrm{err}}$ is an optimistic estimate of $\mathrm{Err}_{\mathcal{T}}$.

Equation (7) indicates that $\mathrm{Err}_{\mathcal{T}}$ can be understood as *extra-sample* error since the test regressor vector $\boldsymbol{x}'$ may differ from the training regressor vector $\boldsymbol{x}$. We can use the *in-sample* error $\mathrm{Err}_{\mathrm{in}}$ instead to illustrate the optimism in $\overline{\mathrm{err}}$

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{N} \sum_{i=1}^{N} E_{\boldsymbol{y}'}[L(y_i', \hat{f}(\boldsymbol{x}_i))|\mathcal{T}], \quad (10)$$

where y_i' denotes new response values at each training point $\boldsymbol{x}_i, i = 1, \ldots, N$. The optimism is defined as the difference between $\mathrm{Err}_{\mathrm{in}}$ and $\overline{\mathrm{err}}$

$$\mathrm{op} = \mathrm{Err}_{\mathrm{in}} - \overline{\mathrm{err}}, \quad (11)$$

which is normally positive. With the fixed training set, the average optimism is computed over the response values in the training set

$$\omega = E_{\boldsymbol{y}}(\mathrm{op}). \quad (12)$$

For squared-error loss functions

$$\omega = \frac{2}{N} \sum_{i=1}^{N} \mathrm{Cov}(y_i, \hat{y}_i). \quad (13)$$

This equation implies that $\overline{\mathrm{err}}$ underestimates the true error by an amount that depends on how closely y_i is correlated with its own predicted value. The tighter the model fits the training data, the greater the covariance will be, thus exaggerating the optimism of $\overline{\mathrm{err}}$. For a linear model of the form $y = f(\boldsymbol{x}) + \varepsilon$

with d independent variables, it can be shown that $\mathrm{Cov}(y_i, \hat{y}_i) = d\sigma_\varepsilon^2$.

From equations (11)–(13), we obtain the following relation for a linear model

$$E_{\boldsymbol{y}}\left(\mathrm{Err}_{\mathrm{in}}\right) = E_{\boldsymbol{y}}\left(\overline{\mathrm{err}}\right) + 2\frac{d}{N}\sigma_\varepsilon^2. \quad (14)$$

The optimism increases with the number of regressors and decreases with the training sample size. Equation (14) suggests that we can estimate the optimism and add it to the training error $\overline{\mathrm{err}}$ to estimate prediction error. This is how Mallows's C_p, AIC, and BIC work.

The C_p statistic for d regressors in a linear model is defined as

$$C_p = \overline{\mathrm{err}} + 2\frac{d}{N}\hat{\sigma}_\varepsilon^2, \quad (15)$$

where $\hat{\sigma}_\varepsilon^2$ is an estimate of the noise variance in a model containing all regressors. The training error is adjusted by a factor proportional to the number of regressors.

The Akaike Information Criterion (AIC), formulated by Akaike [2], is often used when the loss function takes the form of a log-likelihood. As $N \to \infty$, it holds asymptotically that

$$-2E\left[\log \mathrm{Pr}_{\hat{\theta}}(y)\right] \approx -\frac{2}{N}E\left[\log(\hat{\mathcal{L}})\right] + 2\frac{d}{N}, \quad (16)$$

where $\mathrm{Pr}_\theta(y)$ is a set of densities for y, $\hat{\theta}$ is the maximum likelihood estimate of θ, and $\hat{\mathcal{L}}$ is the maximum value of the likelihood function for the model. Thus $\log(\hat{\mathcal{L}}) = \sum_{i=1}^{N} \log \mathrm{Pr}_{\hat{\theta}}(y_i)$. For a Gaussian model with $\sigma_\varepsilon^2 = \hat{\sigma}_\varepsilon^2$ assumed known, $-2\log(\hat{\mathcal{L}}) = \frac{N}{\sigma_\varepsilon^2}\overline{\mathrm{err}}$. AIC defined as

$$\mathrm{AIC} = -2\log(\hat{\mathcal{L}}) + 2d = \frac{N}{\sigma_\varepsilon^2}\left(\overline{\mathrm{err}} + 2\frac{d}{N}\sigma_\varepsilon^2\right) \quad (17)$$

is equivalent to C_p. Given a set of linear models $f_\lambda(\boldsymbol{x})$ indexed by a tuning parameter[b] λ, with associated training error $\overline{\mathrm{err}}(\lambda)$ and $d(\lambda)$ parameters, we choose the tuning parameter $\hat{\lambda}_{\mathrm{AIC}}$ that minimizes

$$\mathrm{AIC}(\lambda) = \frac{N}{\sigma_\varepsilon^2}\left(\overline{\mathrm{err}}(\lambda) + 2\frac{d(\lambda)}{N}\sigma_\varepsilon^2\right). \quad (18)$$

The Bayesian Information Criterion (BIC), also known as the Schwarz Information Criterion (see [17]), is similar to AIC, but with a different penalty for the number of parameters

$$\begin{aligned}\mathrm{BIC} &= -2\log(\hat{\mathcal{L}}) + (\log N)d \\ &= \frac{N}{\sigma_\varepsilon^2}\left[\overline{\mathrm{err}} + (\log N)\frac{d}{N}\sigma_\varepsilon^2\right], \end{aligned} \quad (19)$$

where the factor 2 is replaced by $\log N$. For $N > e^2$, BIC penalizes complex models more heavily than AIC. Like AIC, the chosen model has the tuning parameter that minimizes BIC.

It is not clear whether AIC or BIC performs better, though the latter criterion is asymptotically consistent. Burnham and Anderson [5], Vrieze [21], and Aho et al. [1] show that given a set of candidate models that includes the "true model", as $N \to \infty$, BIC will select the "true model" with probability 1, but AIC tends to choose more complex models. However, when N is small, BIC often selects too simple models. For

[b] Although λ is a continuous parameter, it is usually not feasible to consider all possible values of λ. So we normally discretize its range into a discrete set $\{\lambda_1, \ldots, \lambda_M\}$.

nonlinear and complex models, d should be replaced by some measure of model complexity.

4 CROSS-VALIDATION

Cross-validation is probably the most intuitive and frequently used way to estimate prediction error. It directly estimates the expected extra-sample error $\text{Err} = E[L(y, \hat{f}(\boldsymbol{x}))]$ where the model $\hat{f}(\boldsymbol{x})$ is applied to an independent test set that is not used in estimating the model itself (see [19]). If the data are rich, we can reserve a test set and use it only to assess the predictive ability of the model. Because this is often not the case, cross-validation involves partitioning the available data into subsets, fitting the model on one subset, and testing it on the other subset.

The simplest kind of cross-validation is holdout method. The data are randomly divided into two sets, called the training set and the test set. The test set is typically smaller than the training set. Estimation methods optimize model parameters subject to different values of the tuning parameter λ such that regularized models fit the training set as well as possible. Then the models are asked to predict the response values for the data in the test set. The test errors are used to choose the best model. This evaluation method may be misleading because the test errors can have high variance, depending on how the division is made.

One way to improve over the holdout method is K-fold cross-validation. The data are split into K sets or "folds" of roughly equal size, commonly $K = 5$ or $K = 10$. The holdout method is repeated K times. Each time, a kth set is retained as the test set, and the remaining $K - 1$ sets form the training set to which we fit the model $\hat{f}^{-k}(\boldsymbol{x})$ which is then validated on all of the observations in the kth set. The cross-validation estimate of prediction error is the averaged error across all K trials

$$\text{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}^{-k}(\boldsymbol{x}_i))^2. \quad (20)$$

Given a set of models $f_\lambda(\boldsymbol{x})$ indexed by a tuning parameter λ, we find the tuning parameter $\hat{\lambda}_{\text{CV}}$ that minimizes

$$\text{CV}(\hat{f}_\lambda) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}_\lambda^{-k}(\boldsymbol{x}_i))^2. \quad (21)$$

The advantage of this method is that how the data are divided does not matter much. Each observation belongs to a test set exactly once, and to a training set $K - 1$ times. When $K = 5$ or 10, the cross-validation estimate has low variance, but is potentially an upward biased estimate of the expected prediction error Err if for each fold there are not sufficient training data to fit a good model. When $K = N$, called leave-one-out cross-validation, the cross-validation estimate is approximately unbiased for the expected prediction error Err, but can have high variance because the N training sets are almost identical. In addition, the computational task is rather heavy with N trials. Breiman and Spector [4] and Kohavi [16] recommended 5- or 10-fold cross-validation.

Since the selected $\hat{\lambda}_{\text{CV}}$ that minimizes the cross-validation estimate comes from a random division of the data, its position may be unstable. Small changes in the random-number seed may cause large changes in $\hat{\lambda}_{\text{CV}}$. Breiman et al. [3] suggested that the one standard-error rule should be used to reduce the instability and to choose the most parsimonious model whose error is within one standard error above the error of the minimum.

Another way to secure a more parsimonious model than that based on $\hat{\lambda}_{\text{CV}}$ is adaptive lasso which was introduced by Zou [23]. It is a sequence of cross-validated lassos. After using cross-validation to select a set of nonzero covariates, one constructs coefficient level weights for selected covariates of the form $w_j = \dfrac{1}{|\hat{\beta}_j|^{\delta}}$, where δ is the power to which the weights are raised. Each step chooses either the same covariates chosen by the previous step or a smaller set of them.

5 BOOTSTRAP

The bootstrap was introduced by Efron [6] and was fully developed by Efron and Tibshirani [8]. The bootstrap relies on random sampling with replacement to estimate measures of accuracy of an estimator such as its bias, variance, prediction error, etc. Given a training set of size N, a bootstrap sample is created by drawing N observations randomly with replacement from the training set. This is done B times, yielding B bootstrap samples. For our current purpose, one approach is to fit the model to each of the bootstrap samples and calculate the predicted values for all observations in the original training set. Let $\hat{f}^b(\boldsymbol{x})$ be the fitted model to the bth bootstrap sample, and $\hat{f}^b(\boldsymbol{x}_i), i = 1, \ldots, N$ be the predicted values. The simple bootstrap (SB) estimator of the expected prediction error (see [8]) is

$$\widehat{\text{Err}}_{\text{SB}} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^b(\boldsymbol{x}_i)). \tag{22}$$

But this is not a good estimator because there are overlapping data between the bootstrap samples (which serve as the training sets) and the original training set (which serves as the test set). We can improve it by mimicking cross-validation in which the training and test sets do not have observations in common.

For each observation, we just keep its predicted values from bootstrap samples that do not include that observation. The leave-one-out bootstrap estimate of the expected prediction error (see [7]) is

$$\widehat{\text{Err}}_{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(y_i, \hat{f}^b(\boldsymbol{x}_i)), \tag{23}$$

where C^{-i} is the set of bootstrap samples that do not include observaton i, and $|C^{-i}|$ is the number of these samples. The leave-one-out bootstrap avoids the overfitting problem of $\widehat{\text{Err}}_{\text{SB}}$ but is subject to an upward bias due to the *distance* between the observation i and these bootstrap samples in terms of probability (see [7]). As the observation i has zero probability of belonging

to the bootstrap samples which act as the training sets, we are testing on a set (which includes only observation i) far from the training ones.

Under random sampling with replacement, the probability of an observation not being chosen in a bootstrap sample is $\left(1-\frac{1}{N}\right)^N \approx e^{-1} \approx 0.368$. The probability of an observation being chosen at least once in a bootstrap sample is $1-\left(1-\frac{1}{N}\right)^N \approx 0.632$. Therefore, the effective number of distinct observations contributing to a bootstrap replicate is on average approximately $0.632 \cdot N$. Efron [7] showed that:

$$E_{\boldsymbol{x}}\left[\text{Err}-\overline{\text{err}}\right] \approx 0.632 \cdot E_{\boldsymbol{x}}\left[\widehat{\text{Err}}_{(1)}-\overline{\text{err}}\right]. \tag{24}$$

The 0.632 estimator is

$$\widehat{\text{Err}}_{(0.632)} = 0.368 \cdot \overline{\text{err}} + 0.632 \cdot \widehat{\text{Err}}_{(1)}. \tag{25}$$

It can be understood as a compromise between the training error $\overline{\text{err}}$, which has a downward bias, and the leave-one-out bootstrap estimator. However, the 0.632 estimator does not work well for over-trained binary classifiers (see [4], [9]).

6 AN EMPIRICAL EXAMPLE

In a study of using smoothing splines to refine explanatory modeling, Le [15] revised the housing value equation in Harrison and Rubinfeld [11], who utilized data for 506 census tracts in the Boston Standard Metropolitan Statistical Area (SMSA) in 1970. Variables are defined in Table 1. House prices are certainly determined by house characteristics such as number of rooms, year of construction, property tax rate, distance to employment centers, and index of accessibility to radial highways. Although the neighborhood affects house prices, we do not know for sure which attributes influence property prices and whether they have separate or joint impacts. In stead of using p-values to select neighborhood covariates and interaction terms among house and neighborhood attributes, we now use cross-validated lasso with one standard-error rule to choose the tuning parameter λ and its associated predictors.

Following Le [15], we create a dummy variable, Tax600 which equals 1 for 137 tracts having exceptionally high tax rates exceeding \$600 per \$10,000 and equals 0 otherwise. Stata 16 has built-in commands that make it easy to run lasso. Commands are shown in the Appendix. The one standard-error rule selects $\hat{\lambda}_{\text{SE}} = 0.0025553$. The plot of the cross-validation function shows that $\hat{\lambda}_{\text{SE}}$ is marginally higher than $\hat{\lambda}_{\text{CV}}$, but the model becomes much sparser as forty variables are removed.

Table 1. Variable Definitions

Variable name	**Variable code**	**Description**
Medianvalue	`medv`	Median value of owner-occupied homes.
Room	`rm`	Average number of rooms in owner-occupied homes.
Age	`age`	Proportion of owner-occupied homes built before 1940.
Distance	`dis`	Weighted distance to 5 employment centers in the Boston area.
Highway	`rad`	Highway access index.
Tax	`tax`	Full value property tax rate ($/$10,000).
Black	`black`	Black proportion of population in the community.
Lowstatus	`lstat`	Proportion of population that is lower status = $\frac{1}{2}$(proportion of adults without some high school education and proportion of male workers classified as laborers).
Crime	`crim`	Crime rate by town.
Zoning	`zn`	Proportion of a town's residential land zoned for lots greater than 25,000 square feet.
Industry	`indus`	Proportion of nonretail business acres per town.
Pupil/Teacher	`ptratio`	Pupil-teacher ratio by town school district.
Charles	`chas`	Charles River dummy equals 1 if tract bounds the Charles River and 0 otherwise.
Nox	`nox`	Annual average nitrogen oxide concentration in pphm.

Source: Le (2018).

The penalized coefficients of the twenty five selected covariates are presented in the fourth column of Table 2. Although our purpose is using lasso for prediction, it results have some interesting implications for inference. Variables Room^2, Ln(Distance), Highway, Ln(Tax), Tax600, Crime^2, Room×Lowstatus, Highway×Lowstatus have the same signs as in Le's (2018) model. However, the Age variable now has an expectedly negative impact on house prices. If we run OLS on those twenty five lasso-selected variables, the Age variable is significant. It seems that lasso helps solve the multicollinearity problem since it gives the "right" sign to the Age coefficient.

7 CONCLUSION

Ridge regression and lasso have become popular because they can keep coefficient estimates from blowing up due to multicollinearity, introduce a little bias in return for a considerable decrease in the variance of the predicted values, thus increasing prediction accuracy. In addition, they provide unique solutions for high-dimensional data. However, while ridge-estimated coefficients, although being shrunk, are never zero, lasso sets some coefficients exactly equal to zero, thus gaining more popularity.

It is important to choose an appropriate value of the tuning parameter which governs the amount of penalty placed on the coefficient vector. We can use AIC or BIC to select λ for linear predictive models. BIC works better for very large sample sizes, but often picks up too simple models for small samples. Cross-validation is the simplest and most widely used method to directly estimate prediction error for general models. Five- or ten-fold cross-validation is recommended as they have low variance and relatively low bias in large training sets. Using the same technique in cross-validation and some modification, the 0.632 bootstrap estimator can work well in "light fitting" cases.

ACKNOWLEDGMENTS

I am very grateful to Dr. Hung T. Nguyen for his valuable suggestions in this paper. All errors therein are mine.

APPENDIX

This section presents commands in Stata 16 to run lasso with Harrison and Rubinfeld's [11] data. First of all, we create interaction terms among variables `rm`, `age`, `lndis`, `rad`, `lntax`, `Tax600`, `ptratio`, `black`, `lstat`, `crim`, `zn`, `indus`, `chas`, and `nox`.

We put variables `rm`, `age`, `lndis`, `rad`, `lntax`, `Tax600` in parentheses to make sure that they are in the model regardless of whether lasso wants to select them or not. The `selection(cv, serule)` option specifies the one standard-error rule

Table 2. Housing Value Models

	HR's (1978) Model[a]	Le's (2018) Model	Lasso Model
Constant	4.558	4.595	4.5616
$Room^2$	0.0063	0.0231	0.0261
Age	0.0001	0.0052	-0.0012
Ln(Distance)	-0.1913	-0.2382	-0.2913
Ln(Highway)	0.0957		
Highway		0.0139	0.0462
Tax	-0.0004		
Ln(Tax)		-0.1963	-0.1893
Tax600		-0.3011	-0.1671
Pupil/Teacher	-0.0311	-0.0316	
$(Black - 0.63)^2$	0.3637	0.1240	
Ln(Lowstatus)	-0.3712		
Lowstatus		0.0547	
Crime	-0.0119	-0.0225	
$Crime^2$		0.0002	0.0001
Zoning	0.0001	0.0004	
Industry	0.0002	-0.0074	
Charles	0.0914	0.0474	
Nox^2	-0.6380	-0.7763	
Room×Age		-0.0009	
Room×Lowstatus		-0.0118	-0.0040
Highway×Lowstatus		-0.0012	-0.0006
Highway×Industry		0.0019	
Room×Highway			-0.0027
Room×Pupil/Teacher			-0.0050
Room×$(Black - 0.63)^2$			0.0605
Room×Zoning			0.00002
Room×Nox			-0.1321
Ln(Distance)×Lowstatus			0.0056
Ln(Distance)×Crime			-0.0041
Ln(Distance)×Charles			0.0217
Highway×Zoning			-0.000003
Pupil/Teacher×Zoning			0.0001
$(Black - 0.63)^2$×Crime			-0.0158
$(Black - 0.63)^2$×Industry			0.0036
Crime×Charles			0.0361
Crime×Nox			-0.0143
Zoning×Industry			-0.0002
Charles×Nox			-0.0317
R^2	0.806	0.855	0.861

[a] Results are slightly different from those in Harrison and Rubinfeld (1978).

Source: Author's calculation.

cross-validation to select λ. In addition, we set the `rseed()` option for reproducibility.

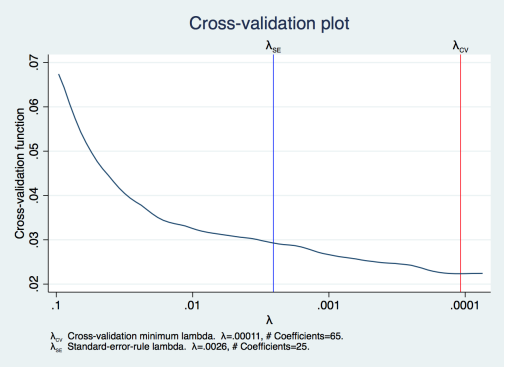
```
. lasso linear lnmedv (rm2 age lndis rad lntax tax600) ptratio black lstat
> crim crim2 zn indus chas nox2 rmage rmlndis rmrad rmlntax rmtax600
> rmptratio rmblack rmlstat rmcrim rmzn rmindus rmchas rmnox agelndis agerad
> agelntax agetax600 ageptratio ageblack agelstat agecrim agezn ageindus
> agechas agenox lndisrad lndislntax lndistax600 lndisptratio lndisblack
> lndislstat lndiscrim lndiszn lndisindus lndischas lndisnox radlntax
> radtax600 radptratio radblack radlstat radcrim radzn radindus radchas
> radnox ptratioblack ptratiolstat ptratiocrim ptratiozn ptratioindus
> ptratiochas ptrationox blacklstat blackcrim blackzn blackindus blackchas
> blacknox lstatcrim lstatzn lstatindus lstatchas lstatnox crimzn crimin-
dus
> crimchas crimnox znindus znchas znnox induschas indusnox chasnox,
> selection(cv, serule) rseed(12345)  10-fold cross-validation with 100 lambdas
...
Grid value 1:        lambda = .0962052      no. of nonzero coef. = 6
Folds: 1...5....10      CVF = .067413
(output omitted)
Grid value 78:       lambda = .0000745      no. of nonzero coef. = 73
Folds: 1...5....10      CVF = .0224233
... cross-validation complete ... minimum found
Lasso linear model                          No. of obs          =        506
                                            No. of covariates   =         89
Selection: Cross-validation one s.e. rule   No. of CV folds     =         10
```

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	.0962052	6	0.5957	.067413
39	lambda before	.0028044	23	0.8229	.0295269
* 40	selected lambda	.0025553	25	0.8244	.0292789
41	lambda after	.0023283	25	0.8257	.0290579
78	last lambda	.0000745	73	0.8655	.0224233

* lambda selected by cross-validation one s.e. rule.

We plot the CV function with the `seline` option that shows the minimum cross-validation $\hat{\lambda}_{\text{CV}}$ and the $\hat{\lambda}_{\text{SE}}$ selected by the one standard-error rule.

```
. cvplot, seline
```



Then, we list the penalized coefficients of the unstandardized variables.

```
. lassocoef, display(coef, penalized)
```

	active
rm2	.0261342
age	-.0011676
lndis	-.2913293
rad	.0462094
lntax	-.1893063
tax600	-.1670989
crim2	.0000688
rmrad	-.0027385
rmptratio	-.0050196
rmblack	.0604683
rmlstat	-.0039817
rmzn	.0000232
rmnox	-.1320919
lndislstat	.0055954
lndiscrim	-.0041265
lndischas	.0217131
radlstat	-.0006211
radzn	-3.12e-06
ptratiozn	.0000577
blackcrim	-.0158256
blackindus	.0035727
crimchas	.0360905
crimnox	-.0143211
znindus	-.0002177
chasnox	-.03174
_cons	4.561571

Legend:
b - base level
e - empty cell
o - omitted

References

[1] Aho, K., Derryberry, D., & Peterson, T. (2014). Model Selection for Ecologists: the Worldviews of AIC and BIC. *Ecology*, 95(3), 631–636.

[2] Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723.

[3] Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees,* Chapman and Hall.

[4] Breiman, L., & Spector, P. (1992). Submodel Selection and Evaluation in Regression: The X-Random Case. *International Statistical Review*, 60(3), 291–319.

[5] Burnham, K. P., & Anderson, D. R. (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd ed., Springer-Verlag.

[6] Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1), 1–26.

[7] Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association,* 78, 316–331.

[8] Efron, B., & Tibshirani, R. (1993). *An Introduction to the Bootstrap,* Chapman & Hall/CRC.

[9] Efron, B., & Tibshirani, R. (1997). Improvements on Cross-Validation: the .632+ Bootstrap Method. *Journal of the American Statistical Association,* 92, 548–560.

[10] Greene, W. H. (2017). *Econometric Analysis*, 8th ed., Pearson.

[11] Harrison, D., & Rubinfeld, D. L. (1978). Hedonic Housing Prices and the Demand for Clean Air. *Journal of Environmental Economics and Management*, 5(1), 81–102.

[12] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer-Verlag.

[13] Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*, 1st ed., Chapman and Hall/CRC.

[14] Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1), 55–67.

[15] Le, C. V. (2018). Smoothing Spline as a Guide to Elaborate Explanatory Modeling. in Kreinovich V., Sriboonchitta S., Chakpitak N. (eds) *Predictive Econometrics and Big Data*, 146–156, Springer.

[16] Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* Volume 2, 1137–1143.

[17] Schwarz, G. E. (1978). Estimating the Dimension of a Model. *Annals of Statistics,* 6(2), 461–464.

[18] Stigler, S. M. (1981). Gauss and the Invention of Least Squares. *The Annals of Statistics*, 9(3), 465–474.

[19] Stone, M. (1977). Asymptotics for and against Cross-Validation. *Biometrika*, 64(1), 29–35.

[20] Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1), 267–288.

[21] Vrieze, S. I. (2012). Model Selection and Psychological Theory: A Discussion of the Differences between the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). *Psychological Methods*, 17(2), 228–243.

[22] Wasserstein, R. L., Schirm, A. L., & Lazar, N. A. (2019). Moving to a World Beyond "$p < 0.05$". *The American Statistician*, 73(S1), 1–19.

[23] Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101, 1418–1429.