

From Nonparametrics to Machine Learning

Zhentaο Shi

April 15, 2020

From Nonparametrics to Machine Learning

公输子削竹木以为鹊，成而飞之，三日不下。公输子自以为至巧。子墨子谓公输子曰：“子之为鹊也，不如匠之为车辖，须臾刘三寸之木，而任五十石之重。”故所为功，利于人谓之巧，不利于人谓之拙。

```
library(magrittr)
set.seed(2020 - 4 - 15)
```

Machine learning has quickly grown into a big field, with applications from scientific research to daily life. An authoritative reference is Friedman, Hastie, and Tibshirani (2008), written at the entry-year postgraduate level. The ideas in machine learning are general and applicable to economic investigation. Athey (2018) discusses the impact of machine learning techniques to economic analysis. Mullainathan and Spiess (2017) survey a few new commonly used methods and demonstrate them in a real example. Taddy (2018) introduces new technology *artificial intelligence* and the implication of the underlying economic modeling.

The two broad classes of machine learning methods are *supervised learning* and *unsupervised learning*. Roughly speaking, the former is about the connection between X and Y , while the latter is only about X . Instances of the former are various regression and classification methods; those of the latter are density estimation, principal component analysis, and clustering. These examples are all familiar econometric problems.

From an econometrician’s view, supervised machine learning is a set of data fitting procedures that focus on out-of-sample prediction. The simplest illustration is in the regression context. We repeat a scientific experiment for n times, and we harvest a dataset $(y_i, x_i)_{i=1}^n$. What would be the best way to predict y_{n+1} from the same experiment if we know x_{n+1} ?

Machine learning is a paradigm shift against conventional statistics. When a statistician propose a new estimator, the standard practice is to pursue three desirable properties one after another. We first establish its consistency, which is seen as the bottom line. Given consistency, we want to show its asymptotic distribution. Ideally, the asymptotic distribution is normal. Asymptotic normality is desirable as it holds for many regular estimators and the inferential procedure is familiar to applied researchers. Furthermore, for an asymptotically normal estimator, we want to show efficiency, an optimality property. An efficient estimator achieves the smallest asymptotic variance in a class of asymptotically normal estimators.

In addition, econometrician also cares about model identification and economic interpretation of the empirical results. Econometrics workflow interacts the data at hand and the model of interest. At the population level, we think about the problem of identification. Once the parameter of interest is identified, then we can proceed to parameter estimation and inference. Finally, we interpret the results and hopefully they shed light on economics.

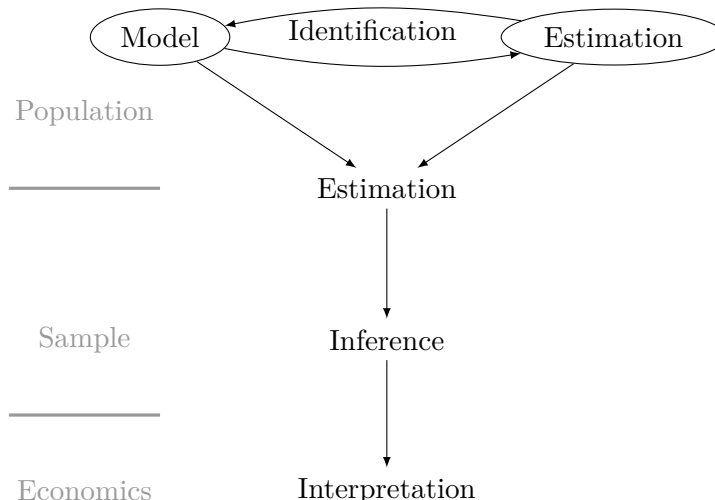


图 1: Econometrics workflow

Machine learning deviates from such routines. First, they argue efficiency is not crucial because the dataset itself is big enough so that the variance is usually small. Second, in many situations statistical inference is not the goal, so inferential procedure is not of interest. For example, the recommendation system on Amazon or Taobao has a machine learning algorithm behind it. There we care about the prediction accuracy, not the causal link why a consumer interested in one good is likely to purchase another good. Third, the world is so complex that we have little idea about how the data is generated. We do not have to assume a data generating process (DGP). If there is no DGP, we lose the standing ground to talk about consistency. Where would my estimator converge to if there is no “true parameter”? With these arguments, the paradigm of conventional statistics is smashed. In the context of econometrics, such argument completely rejects the structural modeling tradition (the Cowles approach).

Readers interested in the debate are referred to Breiman (2001). In this lecture, we put aside the ongoing philosophical debate. Instead, we study the most popular machine learning methods that have found growing popularity in economics.

Nonparametric Estimation

Parametric is referred to problems with a finite number of parameters, whereas *nonparametric* is associated with an infinite number of parameters. Nonparametric estimation is nothing new to statisticians. However, some ideas in this old topic is directly related to the underlying principles of machine learning methods.

Consider the density estimation given a sample (x_1, \dots, x_n) . If we assume that the sample is drawn from a parametric family, for example the normal distribution, then we can use the maximum likelihood estimation to learn the mean and the variance. Nevertheless, when the parametric family is misspecified, the MLE estimation is inconsistent in theory, and we can at best identify a *pseudo true value*. In practice, what is the correct parametric family is unknown. If we do not want to impose a parametric assumption, then in principle we will have to use an infinite number of parameters to fully characterize the density. One well-known nonparametric estimation is the histogram. The shape of the bars of the histogram depends on the partition of the support. If the grid system on the support is too fine, then each bin will have only a few observations. Despite small bias, the estimation will suffer a large variance. On the other hand, if the grid system is too

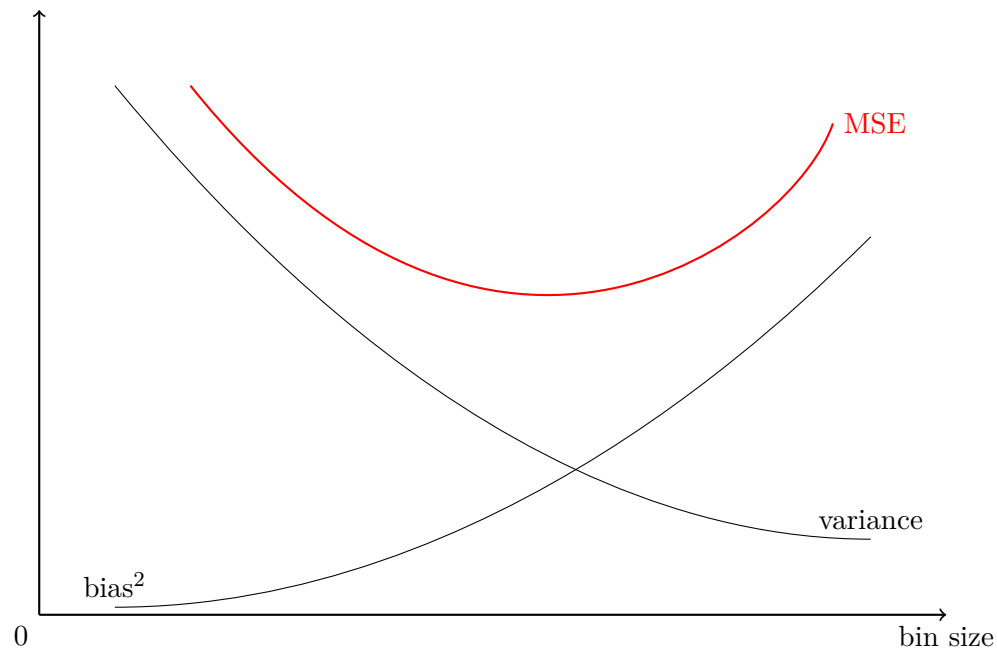


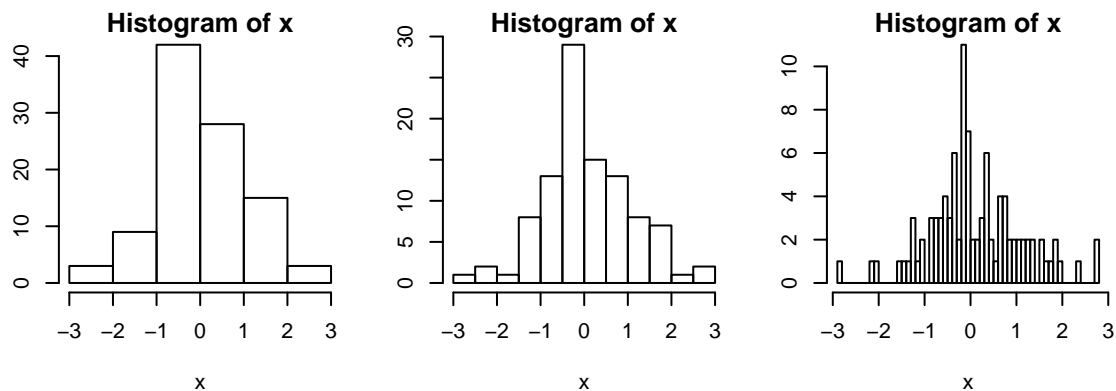
图 2: Bias-variance tradeoff

coarse, then each bin will be wide. It causes big bias, though the variance is small because each bin contains many observations. There is an bias-variance tradeoff. This tradeoff is the defining feature not only for nonparametric estimation but for all machine learning methods.

```
n <- 100
x <- rnorm(n)

par(mfrow = c(1, 3))
par(mar = c(4, 2, 1, 2))

hist(x, breaks = 5)
hist(x, breaks = 15)
hist(x, breaks = 60)
```



Another example of nonparametric estimation is the conditional mean $f(x) = E[y_i | x_i = x]$ given a sample (y_i, x_i) . This is what we encountered in the first lecture of graduate econometrics

Econ5121A. We solve the minimization problem

$$\min_f E[(y_i - f(x_i))^2]$$

In Econ5121A, we use the linear projection to approximate $f(x)$. But the conditional mean is in general a nonlinear function. If we do not know the underlying parametric estimation of (y_i, x_i) , estimating $f(x)$ becomes a non-parametric problem. In practice, the sample size n is always finite. The sample minimization problem is

$$\min_f \sum_{i=1}^n (y_i - f(x_i))^2.$$

We still have to restrict the class of functions that we search for the minimizer. If we assume f is a continuous function, one way to estimate it is the kernel method based on density estimation.

An alternative is to use a series expansion to approximate the function. Series expansion generates many additive regressors whose coefficients will be estimated. This is one way to “create” many variables on the right-hand side of a linear regression. For example, any bounded, continuous and differentiable function has a series representation $f(x) = \sum_{k=0}^{\infty} \beta_k \cos(\frac{k}{2}\pi x)$. In finite sample, we choose a finite K , usually much smaller than n , as a cut-off. Asymptotically $K \rightarrow \infty$ as $n \rightarrow \infty$ so that

$$f_K(x) = \sum_{k=0}^K \beta_k \cos\left(\frac{k}{2}\pi x\right) \rightarrow f(x).$$

Similar bias-variance tradeoff appears in this nonparametric regression. If K is too big, f will be too flexible and it can achieve 100% of in-sample R-squared. This is not useful for out-of-sample prediction. Such prediction will have large variance, but small bias. On the other extreme, a very small K will make $f_K(x)$ too rigid to approximate general nonlinear functions. It causes large bias but small variance.

The fundamental statistical mechanism that governs the performance is the bias-variance trade-off. Thus we need *regularization* to balance the two components in the mean-squared error. Choosing the bandwidth is one way of regularization, choosing the terms of series expansion is another way of regularization.

A third way of regularization is to specify a sufficiently large K , and then add a penalty term to control the complexity of the additive series. The optimization problem is

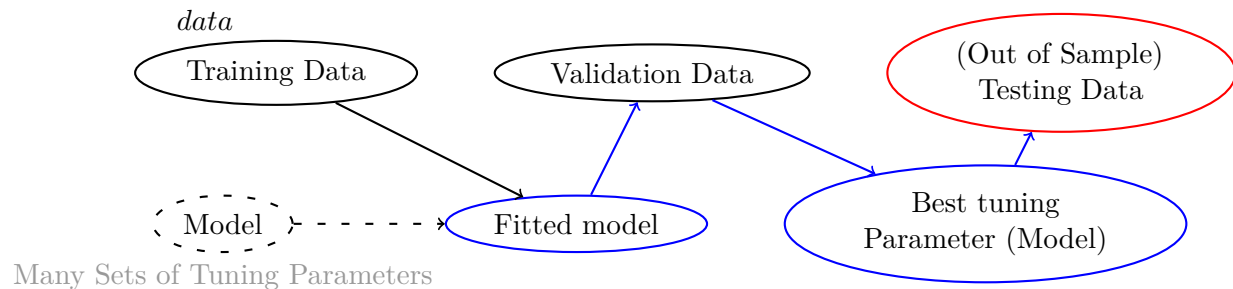
$$\min_{\beta} \frac{1}{2n} \sum_{i=1}^n \left(y_i - \sum_{k=0}^K \beta_k f_k(x_i) \right)^2 + \lambda \sum_{k=0}^K \beta_k^2,$$

where λ is the tuning parameter such that $\lambda \rightarrow 0$ as $n \rightarrow \infty$, and $f_k(x_i) = \cos\left(\frac{k}{2}\pi x_i\right)$. In compact notation, let $y = (y_1, \dots, y_n)'$ and $X = (X_{ik} = f_k(x_i))$, the above problem can be written as

$$(2n)^{-1}(Y - X\beta)'(Y - X\beta) + \lambda \|\beta\|_2^2,$$

and this optimization has an explicit solution $\hat{\beta} = (X'X + \lambda I)^{-1}X'Y$. This is the *ridge regression* proposed in 1970's. This penalization scheme is very similar to what we will discuss in the next section in variable selection.

The practical question is, given a regularization problem, how to choose the tuning parameter? This is a difficult statistical problem with active research. The main theoretical proposal is either



- Data splitting can be done by cross validation
- A data driven approach for feature selection

图 3: Learning workflow

using an *information criterion* (for example, Akaike information criterion $\log \hat{\sigma}^2 + 2K$ or Bayesian information criterion $\log \hat{\sigma}^2 + K \log n$), or *cross validation*.

Data Splitting

The workflow of machine learning methods is quite different from econometrics. The main purpose is often prediction instead of interpretation. They use some “off-the-shelf” generic learning methods, and the models are measured by their performance in prediction. In order to avoid overfitting it is essential to tune at least a few tuning parameters.

Most machine learning methods take an agnostic view about the DGP, and they explicitly acknowledge model uncertainty. To address the issue of model selection (tuning parameter selection), in a data rich environment we split the data into three parts. A *training dataset* is used to fit the model parameter given the tuning parameters. A *validation dataset* is used to compare the out-of-sample performance under different tuning parameters. It helps decide a set of desirable tuning parameters. Ideally, the *testing sample* should be kept by a third party away from the modeler. The testing sample is the final judge of the relative merit of the fitted models.

The R package `caret` (Classification And REgression Training) provides a framework for many machine learning methods. The function `createDataPartition` splits the sample for both cross-sectional data and time series.

Cross Validation

An S -fold cross validation partitions the dataset into S disjoint sections. In each iteration, it picks one of the sections as the validation sample and the other $S - 1$ sections as the training sample, and computes an out-of-sample goodness-of-fit measurement, for example *mean-squared prediction error* $n_v^{-1} \sum_{i \in val} (y_i - \hat{y}_i)^2$ where val is the validation set and n_v is its cardinality, or *mean-absolute prediction error* $n_v^{-1} \sum_{i \in val} |y_i - \hat{y}_i|$. Repeat this process for S times so that each of the S sections are treated as the validation sample, and average the goodness-of-fit measurement over the S sections to determine the best tuning parameter. If $S = n - 1$, it is called *leave-one-out cross validation*, but it can be computationally too expensive when n is big. Instead, in practice we can $S = 5$ for 10, called 5-fold cross validation or 10-fold cross validation, respectively.

Variable Selection and Prediction

In modern scientific analysis, the number of covariates x_i can be enormous. In DNA microarray analysis, we look for association between a symptom and genes. Theory in biology indicates that only a small handful of genes are involved, but it does not pinpoint which ones are the culprits. Variable selection is useful to identify the relevant genes, and then we can think about how to edit the genes to prevent certain diseases and better people's life.

Explanatory variables are abundant in some empirical economic examples. For instance, a questionnaire from the [UK Living Costs and Food Survey](#), a survey widely used for analysis of demand theory and family consumption, consists of thousand of questions. Giannone, Lenza, and Primiceri (2018) [link](#) experiment variable selection methods in 6 widely used economic datasets with many predictors.

Conventionally, applied economists do not appreciate the problem of variable selection, even though they always select variables implicitly. They rely on their prior knowledge to choose variables from a large number of potential candidates. Recently years economists wake up from the long lasting negligence. Stock and Watson (2012) are concerning about forecasting 143 US macroeconomic indicators. They conduct a horse race of several variable selection methods.

The most well-known variable selection method in regression context is the least-absolute-shrinkage-and-selection-operator (Lasso) (Tibshirani 1996). Upon the usual OLS criterion function, Lasso penalizes the L_1 norm of the coefficients. The criterion function of Lasso is written as

$$(2n)^{-1}(Y - X\beta)'(Y - X\beta) + \lambda\|\beta\|_1$$

where $\lambda \geq 0$ is a tuning parameter. Unlike OLS or ridge regression, Lasso does not have a closed-form solution. Fortunately, it is an convex optimization so numerical optimization is fast and reliable for high-dimensional parameter.

In a wide range of values of λ , Lasso can shrink some coefficients exactly to 0, which suggests that these variables are likely to be irrelevant in the regression. However, Zou (2006) finds that Lasso cannot consistently distinguish the relevant variables from the irrelevant ones.

Another successful variable selection method is smoothly-clipped-absolute-deviation (SCAD) (Fan and Li 2001). Its criterion function is

$$(2n)^{-1}(Y - X\beta)'(Y - X\beta) + \sum_{j=1}^d \rho_\lambda(|\beta_j|)$$

where

$$\rho'_\lambda(\theta) = \lambda \left\{ 1\{\theta \leq \lambda\} + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} \cdot 1\{\theta > \lambda\} \right\}$$

for some $a > 2$ and $\theta > 0$. This is a non-convex function, and Fan and Li (2001) establish the so-called *oracle property*. An estimator boasting the oracle property can achieve variable selection consistency and (pointwise) asymptotic normality simultaneously.

The follow-up *adaptive Lasso* (Zou 2006) also enjoys the oracle property. Adaptive Lasso is a two step scheme: 1. First run a Lasso or ridge regression and save the estimator $\hat{\beta}^{(1)}$. 2. Solve

$$(2n)^{-1}(Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^d w_j |\beta_j|$$

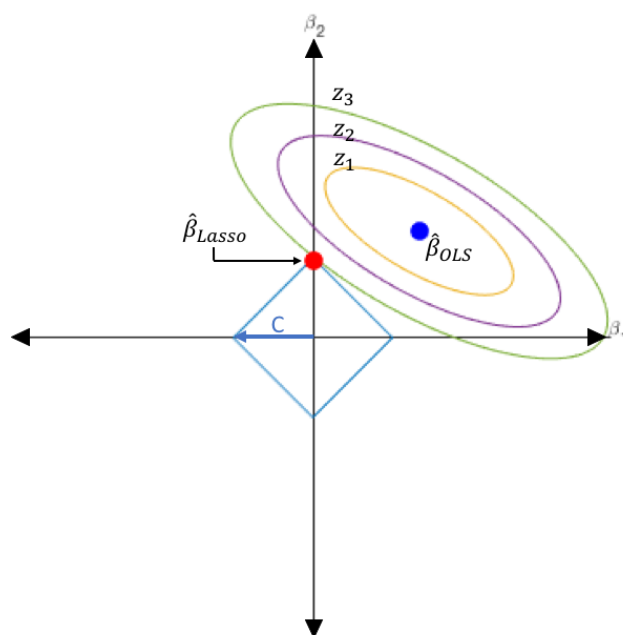


图 4: Lasso with two parameters

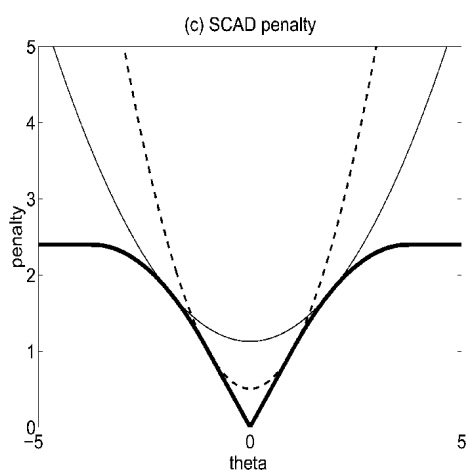


图 5: SCAD criterion and quadratic approximation

where $w_j = 1 / |\hat{\beta}_j^{(1)}|^a$ and $a \geq 1$ is a constant. (Common choice is $a = 1$ or 2).

In R, `glmnet` or `LARS` implements Lasso, and `ncvreg` carries out SCAD. Adaptive Lasso can be done by setting the weight via the argument `penalty.factor` in `glmnet`.

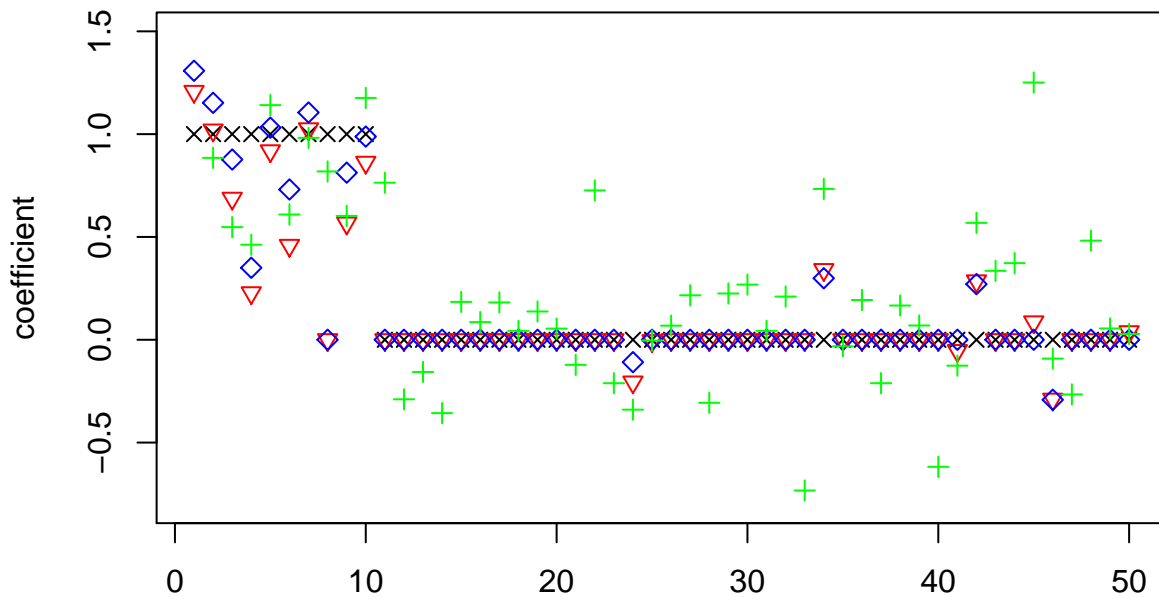
```
n <- 40
p <- 50
b0 <- c(rep(1, 10), rep(0, p - 10))
x <- matrix(rnorm(n * p), n, p)
y <- x %*% b0 + rnorm(n)

ols <- MASS::ginv(t(x) %*% x) %*% (t(x) %*% y) # OLS
# Implement Lasso by glmnet
cv_lasso <- glmnet::cv.glmnet(x, y)
lasso_result <- glmnet::glmnet(x, y, lambda = cv_lasso$lambda.min)

# Get weights
b_temp <- as.numeric(lasso_result$beta)
b_temp[b_temp == 0] <- 1e-8
w <- 1 / abs(b_temp) # Let gamma = 1

# Implement Adaptive Lasso by glmnet
cv_lasso <- glmnet::cv.glmnet(x, y, penalty.factor = w)
alasso_result <-
  glmnet::glmnet(x, y, penalty.factor = w, lambda = cv_lasso$lambda.min)

plot(b0, ylim = c(-0.8, 1.5), pch = 4, xlab = "", ylab = "coefficient")
points(lasso_result$beta, col = "red", pch = 6)
points(alasso_result$beta, col = "blue", pch = 5)
points(ols, col = "green", pch = 3)
```

```
# out of sample prediction
x_new <- matrix(rnorm(n * p), n, p)
y_new <- x_new %*% b0 + rnorm(n)
lasso_msfe <- (y_new - predict(lasso_result, newx = x_new)) %>% var()
alasso_msfe <- (y_new - predict(alasso_result, newx = x_new)) %>% var()
ols_msfe <- (y_new - x_new %*% ols) %>% var()

print(c(lasso_msfe, alasso_msfe, ols_msfe))
```

```
## [1] 2.927166 2.567215 13.063155
```

We can DIY Lasso by CVXR.

```
library(CVXR)

lambda <- 2 * cv_lasso$lambda.min # tuning parameter

# CVXR for Lasso
beta_cvxr <- Variable(p)
obj <- sum_squares(y - x %*% beta_cvxr) / (2 * n) + lambda * p_norm(beta_cvxr, 1)
prob <- Problem(Minimize(obj))
lasso_cvxr <- solve(prob)
beta_cvxr_hat <- lasso_cvxr$getValue(beta_cvxr) %>% as.vector() %>% print()
```

```
## [1] 1.115222e+00 7.364074e-01 4.572155e-01 1.374911e-01 7.496561e-01
## [6] 9.522302e-02 1.020766e+00 4.112035e-11 3.042626e-01 7.233749e-01
## [11] -4.522813e-11 4.218195e-12 -2.691423e-11 6.351772e-11 -6.719435e-12
## [16] 1.200056e-11 1.712818e-11 -6.572958e-12 -3.800858e-11 7.704580e-12
## [21] 9.415925e-13 1.698305e-11 9.302759e-12 -2.177665e-01 -4.777442e-11
## [26] -5.884406e-11 1.140063e-10 1.773145e-11 -6.097165e-14 5.263170e-11
## [31] 4.932108e-11 9.797570e-12 -1.072903e-11 3.343054e-01 3.511021e-11
## [36] 1.473053e-11 -1.344543e-11 3.523223e-11 -5.246665e-11 -2.172122e-10
## [41] -4.917203e-11 1.838269e-01 -4.279397e-12 5.790897e-12 7.478519e-11
## [46] -1.820539e-01 4.632804e-12 -1.756539e-11 1.725078e-11 4.581891e-10
```

More methods are available if prediction of the response variables is the sole purpose of the regression. An intuitive one is called *stagewise forward selection*. We start from an empty model. Given many candidate x_j , in each round we add the regressor that can produce the biggest R^2 . This method is similar to the idea of L_2 componentwise boosting, which does not adjust the coefficients fitted earlier.

Shrinkage Estimation in Econometrics

- Su, Shi, and Phillips (2016): use shrinkage estimation for classification
- Shi (2016): convergence rate of GMM Lasso
- Lee, Shi, and Gao (2018): Lasso and adaptive Lasso in predictive regression
- Shi and Huang (2019): forward selection

Empirical Applications

- Lehrer and Xie (2017): movie box office
- Feng, Giglio, and Xiu (2019): factor zoo, compare machine learning methods
- Chinco, Clark-Joseph, and Ye (2019): financial market, Lasso prediction

Reading

- Efron and Hastie: Ch. 16
- Athey (2018)

Appendix

Suppose $y_i = x_i' \beta_0 + e_i$, where e_i is independent of x_i and $\text{var}[e_i] = \sigma^2$. Then

$$\min_{\beta} E[(y_i - x_i' \beta)^2] = E[(y_i - x_i' \beta_0)^2] = E[e_i^2] = \sigma^2.$$

This is the minimal error that can be achieved in the population.

In reality, we have a sample (y_i, x_i) of n observations, and we estimate β by the OLS estimator $\hat{\beta} = (X'X)^{-1}X'y$. The expectation of the SSR is

$$E \left[\frac{1}{n} \sum_{i=1}^n (y_i - x_i' \hat{\beta})^2 \right] = \frac{1}{n} E \left[e'(I_n - X(X'X)^{-1}X)e \right] = \frac{\sigma^2}{n} (n - p) = \sigma^2 \left(1 - \frac{p}{n} \right) < \sigma^2$$

Asymptotically, if $p/n \rightarrow 0$, the two risks converge. Otherwise if $p/n \rightarrow c$, the expected SSR is strictly smaller than the minimal population risk. The model is overfitted.

References

- Athey, Susan. 2018. “The Impact of Machine Learning on Economics.” In *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press.
- Breiman, Leo. 2001. “Statistical Modeling: The Two Cultures (with Comments and a Rejoinder by the Author).” *Statistical Science* 16 (3): 199–231.
- Chinco, Alexander M, Adam D Clark-Joseph, and Mao Ye. 2019. “Sparse Signals in the Cross-Section of Returns.” *Journal of Finance* forthcoming.
- Fan, Jianqing, and Runze Li. 2001. “Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties.” *Journal of the American Statistical Association* 96 (456): 1348–60.
- Feng, Guanhao, Stefano Giglio, and Dacheng Xiu. 2019. “Taming the Factor Zoo: A Test of New Factors.” *Fama-Miller Working Paper*, 17–04.
- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2008. *The Elements of Statistical Learning*. 2nd ed. Springer.
- Giannone, Domenico, Michele Lenza, and Giorgio E Primiceri. 2018. “Economic Predictions with Big Data: The Illusion of Sparsity.” Federal Reserve Bank of New York Staff Reports.
- Lee, Ji Hyung, Zhentao Shi, and Zhan Gao. 2018. “On Lasso for Predictive Regression.” arXiv: 1810.03140 [econ. EM].
- Lehrer, Steven, and Tian Xie. 2017. “Box Office Buzz: Does Social Media Data Steal the Show from Model Uncertainty When Forecasting for Hollywood?” *Review of Economics and Statistics* 99 (5): 749–55.
- Mullainathan, Sendhil, and Jann Spiess. 2017. “Machine Learning: An Applied Econometric Approach.” *Journal of Economic Perspectives* 31 (2): 87–106.
- Shi, Zhentao. 2016. “Estimation of Sparse Structural Parameters with Many Endogenous Variables.” *Econometric Reviews* 35 (8-10): 1582–1608.
- Shi, Zhentao, and Jingyi Huang. 2019. “Forward-Selected Panel Data Approach for Program Evaluation.” *arXiv Preprint arXiv:1908.05894*.
- Stock, James H, and Mark W Watson. 2012. “Generalized Shrinkage Methods for Forecasting Using Many Predictors.” *Journal of Business & Economic Statistics* 30 (4): 481–93.
- Su, Liangjun, Zhentao Shi, and Peter CB Phillips. 2016. “Identifying Latent Structures in Panel Data.” *Econometrica* 84 (6): 2215–64.
- Taddy, Matt. 2018. “The Technological Elements of Artificial Intelligence.” National Bureau of Economic Research.
- Tibshirani, Robert. 1996. “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–88.
- Zou, Hui. 2006. “The Adaptive Lasso and Its Oracle Properties.” *Journal of the American Statistical Association* 101 (476): 1418–29.