Applied Causal Inference Powered by ML and AI

Victor Chernozhukov* Christian Hansen† Nathan Kallus‡
Martin Spindler§ Vasilis Syrgkanis¶

February 28, 2024

Publisher: Online
Version 0.1.1

* MIT
† Chicago Booth
‡ Cornell University
§ Hamburg University
¶ Stanford University

Predictive Inference via Modern High-Dimensional Linear Regression

3

"Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher."
(perfection is attained not when there is no longer anything to add, but when there is no longer anything to take away.)

– Antoine de Saint-Exupéry [1].

Here we discuss the use of penalized regressions for constructing predictions in high-dimensional settings, particularly when $p > n$. We first motivate the high-dimensional setting as arising both from having a high-dimensional regressor set and from constructing technical regressors from raw regressors. We then discuss Lasso, which penalizes the size of the model by the sum of the absolute value of its coefficients. We conclude with an overview of other penalized regression methods.

3.1 Linear Regression with High-Dimensional Covariates

The Framework

We consider a regression model

$$Y = \beta' X + \epsilon, \quad \epsilon \perp X,$$

where $\beta' X$ is the population best linear predictor of Y using X, or simply the population linear regression function. The vector $X = (X_j)_{j=1}^p$ is p-dimensional. That is, there are p regressors, and

$$p \text{ is large, possibly much larger than } n.$$

This case where p is large relative to the sample size is what we call a *high-dimensional* setting. High-dimensional settings arise when

- data have large dimensional features (i.e. many covariates are available for use as regressors),
- we construct many technical regressors[1] from raw regressors, or
- both.

1: Recall, a *technical regressor* is any variable obtained as a transformation of a basic regressor.

Examples of datasets where many covariates are available and potential corresponding exemplary applications include

- country characteristics in cross-country wealth analysis,
- housing characteristics in house pricing/appraisal analysis,
- individual health information in electronic health records and claims data, and
- product characteristics at the point of purchase in demand analysis.

Another source of high-dimensionality is the use of constructed features or regressors of the form

$$X = T(W) = (T_1(W), ..., T_p(W))',$$

where W denotes original raw regressors. As we discussed in Chapter 1, the set of transformations $T(W)$ is sometimes called the *dictionary* of transformations. Example transformations include polynomials, splines, interactions between variables, and applying functions such as the logarithm or exponential. In the wage analysis in Chapter 1, for example, we used quadratic and cubic transformations of experience, as well as interactions

(products) of these regressors with education and geographic indicators. Recall that the main motivation for the use of constructed regressors is to build *more flexible and potentially better* prediction rules.

The potential for improved prediction arises because we are using prediction rules $\beta'X = \beta'T(W)$ that are *nonlinear* in the original raw regressors W and may thus capture more complex patterns that exist in the data. Conveniently, the prediction rule $\beta'X$ is still linear with respect to the parameters, β, and with respect to the constructed regressors $X = T(W)$, so inherits much from the previous discussion of linear regression provided in Chapter 1.

In summary, we have provided two motivations for using high-dimensional regressors in prediction:

- The first motivation is that modern datasets have high-dimensional features that can be used as regressors.
- The second motivation is that we can use nonlinear transformations of features or raw regressors and their interactions to form constructed regressors. Using transformations allows us to better approximate the best prediction rule – the conditional expectation of the outcome given raw regressors.

Lasso

Recall that we are considering a regression model

$$Y = \beta'X + \epsilon = \sum_{j=1}^{p} \beta_j X_j + \epsilon, \quad \epsilon \perp X \tag{3.1.1}$$

where p is possibly much larger than n.

Classical linear regression or least squares fails in these high-dimensional settings because it *overfits* in finite samples. Intuitively, overfitting refers to using patterns that are idiosyncratic to a specific dataset and do not generalize out of sample. That is, it corresponds to using a prediction rule that is overly complex in that it uses patterns that help explain a given dataset, increasing in-sample measures of fit, but are not present in different data even if the data are drawn from the same population, potentially harming out-of-sample prediction performance.

The potential for classical linear regression estimated with least squares to overfit is especially apparent when $p \geq n$. In this case,

conventional least squares will perfectly fit the data regardless of the value of β as long as the covariate matrix is rank n.[2] We therefore make some assumptions and modify the regression method to deal with cases where p is large.

2: Recall that we illustrated the problem with overfitting in Section 1.2.

An intuitive starting point is the assumption of *approximate sparsity*. Under approximate sparsity, there is a small group of regressors with relatively large coefficients whose use alone suffices to approximate the BLP $\beta'X$ well. The rest of the regressors are assumed to have relatively small coefficients and contribute little to the approximation of the BLP.

An example of approximate sparsity is captured by regression coefficients of the form[3]

$$\beta_j \propto 1/j^2, \quad j = 1, ..., p.$$

3: The notation $\propto$ reads as "proportional to."

Here, the first few coefficients capture almost all the explanatory power of the full vector of coefficients as shown in Figure 3.1.

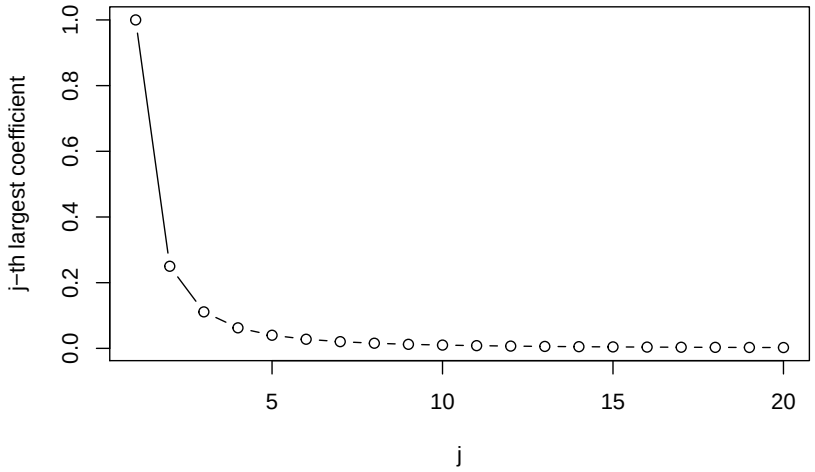

Figure 3.1: Example of regression coefficients, $\beta_j = 1/j^2$ that satisfy approximate sparsity.

Next, we define approximate sparsity formally.

Definition 3.1.1 ***Approximate sparsity:*** *The sorted absolute values of the coefficients decay quickly. Specifically, the j^{th} largest coefficient (in absolute value) denoted by $|\beta|_{(j)}$ obeys*

$$|\beta|_{(j)} \leq Aj^{-a}, \quad a > 1/2, \tag{3.1.2}$$

for each j, where the constants a and A do not depend on the sample size n.

For estimation purposes, we have a random sample $\{(Y_i, X_i)\}_{i=1}^n$. We seek to construct a good linear predictor $\hat{\beta}'X$, which works well when p/n is not small.

Before defining the Lasso problem, it is important to note that we are treating all variables as centered and thus do not include an intercept in the model. In practice, this construction means that, for raw variables Y^* and X^*, we start by defining demeaned versions of these variables $Y = Y^* - \mathbb{E}_n[Y^*]$ and $X = X^* - \mathbb{E}_n[X^*]$ for use in estimation of model parameters.[4] We note that the centered model (3.1.1) is equivalent to starting with the model

$$Y^* = \alpha + \beta' X^* + \epsilon \quad \epsilon \perp X^*$$

with intercept $\alpha = \mathrm{E}[Y^*] - \beta' \mathrm{E}[X^*]$. For estimates $\hat{\beta}$ obtained by estimating (3.1.1), we can thus recover an estimate of α as $\hat{\alpha} = \mathbb{E}_n[Y^*] - \hat{\beta}' \mathbb{E}_n[X^*]$.

A centered random variable U has $\mathrm{E}[U] = 0$, and a centered variable U in a sample has $\mathbb{E}_n[U] = 0$.

4: When performing validation exercises, demeaning and any other transformations that depend on features of the data, such as standardization, should be done in both training and test data using the features of the *training* data rather than of the full sample or the test data.

When discussing theoretical properties, we will further assume that regressors are normalized,

$$\mathrm{E}[X_j^2] = 1.$$

We do state the estimation algorithms without assuming this normalization. The combination of centering and normalization – *standardization* – is commonly employed in practice and is done by default in many software packages.

Lasso constructs $\hat{\beta}$ as the solution of the following penalized least squares problem:

$$\min_{b \in \mathbb{R}^p} \quad \sum_i (Y_i - b' X_i)^2 + \lambda \cdot \sum_{j=1}^{p} |b_j| \hat{\psi}_j, \tag{3.1.3}$$

which is called the Lasso regression problem. The first term is n times the sample mean squared error, and the second term is called a *penalty term*. The penalty term introduces a cost to the complexity of the prospective model where complexity is captured by the sum of the products of the absolute values of the coefficients b_j with the *penalty loadings* $\hat{\psi}_j$ all multiplied by the *penalty level* λ.

Rather than work with centered variables, we could equivalently define (3.1.3) with an intercept where the intercept *does not* enter the penalty function. The important thing to keep in mind is that it is rarely appropriate to penalize the intercept.

The penalty loadings are typically set as

$$\hat{\psi}_j = \sqrt{\mathbb{E}_n[X_j^2]}.$$

The use of this penalty ensures invariance of Lasso predictions to rescaling X_j'. Note that many software packages implement the Lasso with simple penalty loadings $\hat{\psi}_j = 1$. In such cases,

the use of standardized variables produces the same results as using these penalty loadings.

As long as $\lambda > 0$, the introduction of the penalty term in (3.1.3) leads to a prediction rule which is less complex than the rule that would be obtained via solving the unpenalized least squares problem. Specifically, the penalty term in the Lasso problem, $\sum_{j=1}^{p} |b_j| \hat{\psi}_j$, provides a measure of complexity of a regression model in terms of the overall magnitude of the coefficients. When λ is positive, minimizing the Lasso problem requires trading off in-sample fit with this measure of complexity. As a result, the overall magnitude of the estimated coefficients, as measured by the penalty term, will be smaller than the overall magnitude of the coefficients absent this penalty. That is, the Lasso solution will have coefficients that are "shrunk" towards 0 relative to the unpenalized least squares problem.[5]

5: This overall shrinkage towards zero relative to the unpenalized problem is sometimes referred to as *shrinkage bias* or *regularization bias*.

One important benefit of introducing the penalty term is that it helps guard against overfitting by introducing a cost to model complexity. Intuitively, overfitting occurs as a model is made increasingly complex in an effort to make improvements to in-sample fit that are small relative to sampling error and could thus correspond to idiosyncrasies of a specific finite sample. The penalty term imposes a cost to complexity which help keep increases to complexity that have small benefit in terms of improving fit from being made. Through careful choice of λ, we can theoretically guarantee that the Lasso predictor is similar to the optimal predictor, and thus generalizable, even in high-dimensional settings.

A second important feature of Lasso is that it imposes the approximate sparsity condition on the estimated coefficients $\hat{\beta}$. Approximate sparsity is produced because the penalty function in (3.1.3) has a kink at zero which results in the marginal cost of including regressor X_j ($\lambda \hat{\psi}_j > 0$) always being positive when $\lambda > 0$. Therefore, Lasso includes a regressor X_j with non-zero coefficient only if its marginal predictive ability is higher than this marginal cost threshold. That is, Lasso does *variable selection*: The Lasso solution drops any variable (equivalently sets the variable's coefficient to 0) whose marginal predictive benefit does not exceed the marginal cost of inclusion. We illustrate this variable selection property numerically in Example 3.1.1 below.

It is important to note that Lasso will not generally select the "right" set of variables. Lasso will tend to exclude variables with small, but non-zero population coefficients. Lasso will also tend to fail to select the right variables in settings where the

X variables are correlated.[6] That is, one should not conclude that Lasso has selected exactly the variables with non-zero coefficients in the population unless one can rule out variables with small, but non-zero coefficients and ensure that variables are all at most weakly correlated.[7] This failure does not mean that the Lasso predictions are poor quality, but does mean that care should be taken in interpreting the selected variables.

Example 3.1.1 (Simulation Example) Consider

$$Y = \beta' X + \epsilon, \quad X \sim N(0, I_p), \quad \epsilon \sim N(0,1),$$

with approximately sparse regression coefficients:

$$\beta_j = 1/j^2, \quad j = 1, ..., p$$

and

$$n = 300, \quad p = 1000.$$

Figure 3.2 shows that $\hat{\beta}$ is sparse and is close to β. We see that Lasso sets most of regression coefficients to zero. It figures out *approximately* the right set of regressors, including only those with the two largest coefficients. Note that Lasso does not, and in fact cannot, select the regressors with non-zero coefficients in this example as all variables have non-zero coefficients.

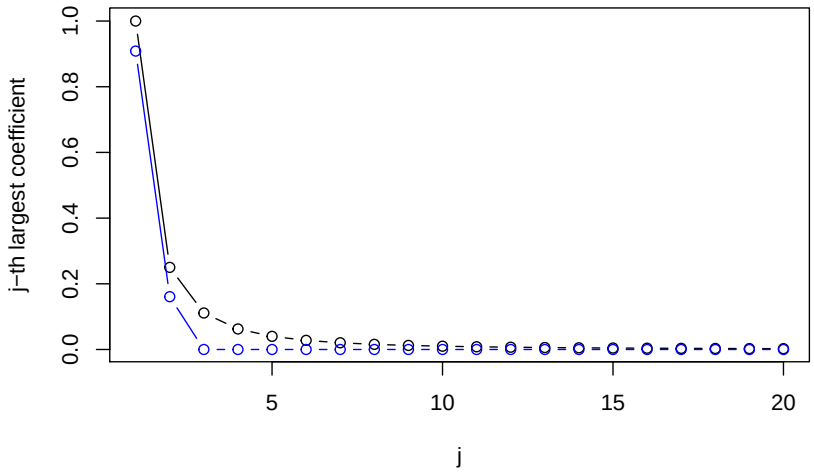

Figure 3.2: The true coefficients (black) vs. coefficients estimated by Lasso (blue) in Example 3.1.1.

A crucial point for the two Lasso properties that we have discussed is the choice of the penalization parameter λ. A theoretically valid choice is[8]

$$\lambda = 2 \cdot c\hat{\sigma}\sqrt{n} z_{1-\mathsf{a}/(2p)} \tag{3.1.4}$$

where $\hat{\sigma} \approx \sigma = \sqrt{\mathrm{E}[\epsilon^2]}$ is obtained via an iteration method defined in Appendix 3.A, $c > 1$, and $1 - \mathsf{a}$ is a confidence

6: For example, consider a scenario where variable X_1 has coefficient $\beta_1 = 0$ but is highly correlated to variables $X_2, ..., X_k$ that have non-zero coefficients. It is quite plausible that the marginal predictive benefit of including X_1 in the model is very high when $X_2, ..., X_k$ are not in the model while the marginal predictive benefit of any one of $X_2, ..., X_k$ is relatively low. In this case, X_1 may enter the Lasso solution with a non-zero coefficient while all of $X_2, .., X_k$ are excluded.

7: This inability to select *exactly* the right regressors is not special to Lasso but shared by all variable selection procedures.

8: Recall that z_t is such that $\mathrm{P}((N(0,1) \leq z_t) = t$.

level.[9] We can further simplify the choice using Feller's tail inequality:

$$z_{1-\mathsf{a}/(2p)} \leq \sqrt{2\log(2p/\mathsf{a})},$$

where the inequality becomes sharp as $p \to \infty$.

This penalty level ensures that the Lasso predictor $\hat{\beta}'X$ does not overfit the data and delivers good predictive performance under approximate sparsity ([2, 3]). Another good way to pick the penalty level when building a model for prediction is by cross-validation ([4]).[10]

9: Practical recommendations, based on theory and that seem to work well in practice, are to set $c = 1.1$ and $\mathsf{a} = .05$.

10: Cross-validation is a repeated data-splitting method for choosing penalty parameters for Lasso and for selecting among predictive models more generally. We outline the basic idea of cross-validation in Section 3.B.

Quick Heuristics for Lasso Properties and Penalty Choice*

Here, we provide a sketch of the mathematics of the Lasso estimator illustrating its variable selection properties and motivating the choice of λ in (3.1.4).

Assume $\hat{\psi}_j = 1$ for simplicity. The j-th component $\hat{\beta}_j$ of the Lasso estimator $\hat{\beta}$ is set to zero if the marginal predictive benefit of changing $\hat{\beta}_j$ away from zero is smaller than the marginal increase in penalty (see Figure 3.3):

$$\hat{\beta}_j = 0 \text{ if } \left| \frac{\partial}{\partial \hat{\beta}_j} \sum_i (Y_i - \hat{\beta}' X_i)^2 \right| < \lambda.$$

That is,

$$\hat{\beta}_j = 0 \text{ if } |-\hat{S}_j| < \lambda, \quad \hat{S}_j = 2\sum_i (Y_i - \hat{\beta}' X_i) X_{ji}.$$

We discuss more detailed heuristics for penalty level selection in the appendix, but the rough idea is that the penalty should dominate the noise $S_j = 2\sum_i (Y_i - \beta' X_i) X_{ji}$ in the measurement of the marginal predictive ability. By the high-dimensional central limit theorem ([5]), we have that

$$(S_j)_{j=1}^p \overset{\mathsf{a}}{\sim} 2\sqrt{n}\sigma (\mathcal{N}_j)_{j=1}^p, \quad \mathcal{N}_j \sim N(0,1).$$

Therefore, to guarantee that Lasso sets to zero the any coefficient whose actual value is zero, we would like to choose λ to dominate

$$2\sqrt{n}\sigma \max_{j=1,\ldots,p} |\mathcal{N}_j|$$

with high probability, say $1 - \mathsf{a}$. Then by the union bound and

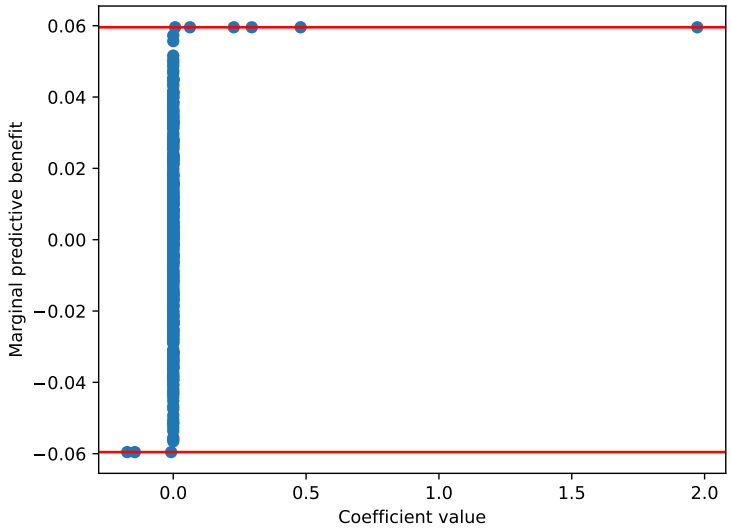

Figure 3.3: Example relationship between coefficient value and (signed) marginal predictive value $\hat{S}_j$ at the optimal solution to the Lasso objective. The red lines correspond to $\{-\lambda, \lambda\}$.

symmetry of centered normal variables,

$$\begin{aligned} &\mathrm{P}\Big(\max_{j=1,\ldots,p} |\mathcal{N}_j| > z_{1-\mathsf{a}/(2p)}\Big) \\ &\leq 2\sum_{j=1}^{p} \mathrm{P}\left(\mathcal{N}_j > z_{1-\mathsf{a}/(2p)}\right) \\ &= 2p\Big(1-(1-\mathsf{a}/(2p))\Big) = \mathsf{a}. \end{aligned}$$

The union bound here is crude, but the bound is not very loose. In particular, when the $\mathcal{N}_j$'s are independent, the bound becomes sharp as $p \to \infty$. Finally, setting

$$\lambda = 2\sigma\sqrt{n} z_{1-\mathsf{a}/(2p)}$$

we conclude that

$$\mathrm{P}(\max_j |S_j| \leq \lambda) \geq 1 - \mathsf{a},$$

up to a vanishing error. That is, this choice of λ guarantees that variables with $\beta_j = 0$ are excluded from the model (have $\hat{\beta}_j = 0$) with high probability.

OLS Post-Lasso

We can use the Lasso-selected set of regressors, those regressors whose Lasso coefficient estimates are non-zero, to refit the model by least squares. This method is called "least squares post Lasso" or simply *Post-Lasso* ([3]). Compared to Lasso,

Post-Lasso undoes the overall shrinkage toward zero relative to unconstrained least squares from the estimated non-zero coefficients, as we illustrate in Figure 3.1.5 below.[11] Removing this shrinkage towards zero from the non-zero coefficients sometimes delivers improvements in predictive performance.

11: Note that the estimates of the large coefficients are nearly perfect after OLS refitting of the model selected by Lasso in this example.

Post-Lasso. We define the Post-Lasso

$$\widetilde{\beta} \in \arg\min_{b \in \mathbb{R}^p} \sum_i (Y_i - b' X_i)^2 \text{ such that } b_j = 0 \text{ if } \hat{\beta}_j = 0 \text{ for each } j, \tag{3.1.5}$$

where $\hat{\beta}$ is the Lasso coefficient estimator. The formal properties of the Post-Lasso estimator $\widetilde{\beta}$ are similar to those of Lasso $\hat{\beta}$; see Section 3.2.

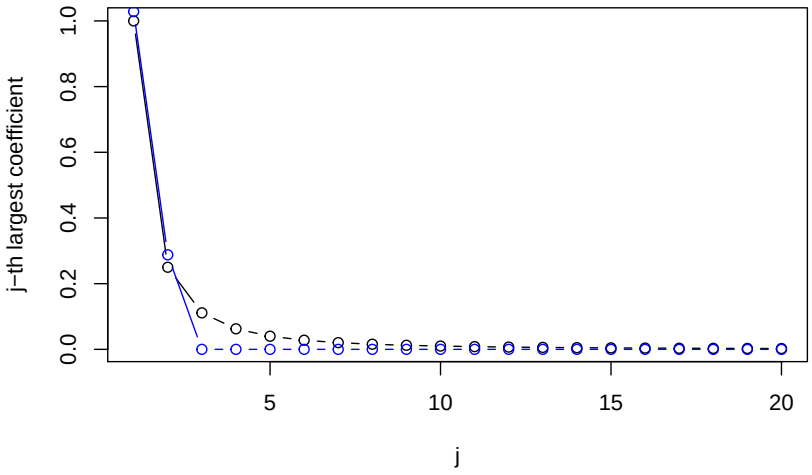

Figure 3.4: The true coefficients (black) vs. coefficients estimated by Post-Lasso (blue) in the Example 3.1.1. Post-Lasso tends to remove regularization bias from the estimated non-zero coefficients.

Remark 3.1.1 (Cross Validation and OLS Post-Lasso) Note that, when using Post-Lasso, one should either use the theoretically justified penalty parameter ([3]) as outlined above or cross-validation for the overall OLS Post-Lasso process. That is, one *should not* apply cross-validation to the Lasso to find a value for λ and then use this same value of λ with Post-Lasso. Unsurprisingly, using a penalty parameter chosen to optimize cross-validation performance for Lasso tends to lead to poor empirical performance when applied to an entirely different procedure, Post-Lasso.

3.2 Predictive Performance of Lasso and Post-Lasso

The best linear prediction rule (out-of-sample) is $\beta'X$. We want to understand the quality of the Lasso prediction rule, $\hat{\beta}'X$. That is,

- Does $\hat{\beta}'X$ provide a good approximation to $\beta'X$?

Recall that with Lasso, we are trying to estimate p parameters $\beta_1, ..., \beta_p$, imposing approximate sparsity via penalization. Under approximate sparsity, only a few, say s, parameters will be "important." We can call s the *effective dimension*. Lasso approximately figures out which parameters are important to keep. Further, intuitively, to estimate each of the "important" s parameters well, we need many observations for each such parameter. This means that n/s must be large, or, equivalently s/n must be small. Using previous reasoning from least squares theory, we might also conjecture that the key determinant of the rate at which Lasso approximates the best linear predictor is $\sqrt{s/n}$. This conjecture is almost correct.

Theorem 3.2.1 *Under approximate sparsity as defined in Definition 3.1.1, restricted isometry conditions stated below, choosing λ as in* (3.1.4)*, and other regularity conditions stated e.g. in [3, 6], with probability approaching $1-\alpha$ as $n \to \infty$, the following bound holds:*

$$\sqrt{\mathrm{E}_X\left[(\beta'X - \hat{\beta}'X)^2\right]} \leq \text{const} \cdot \sqrt{\mathrm{E}[\epsilon^2]}\sqrt{\frac{s\log(\max\{p,n\})}{n}},$$

where E_X denotes expectation with respect to X, and the effective dimension is

$$s = \text{const} \cdot A^{1/a} \cdot n^{\frac{1}{2a}},$$

where constant a is the speed of decay of the sorted coefficient values in the approximate sparsity definition, Definition 3.1.1. Moreover, the number of regressors selected by Lasso is bounded by

$$\text{const} \cdot s$$

with probability approaching $1-\mathsf{a}$ as $n \to \infty$. The constants const *are different in different places and may depend on the distribution of (Y, X) and on* a.

The definition of effective dimension stated in this theorem applies, for instance, under the regularity condition that $\max_{j=1}^{p} \|\mathrm{E}[X_j X]\|_1 \leq \text{const}$; i.e. the sum of the absolute values of every row of the covariance matrix $\mathrm{E}[XX']$ is at most a constant. One can also obtain appropriate notions of effective dimension under weaker assumptions on the covariance matrix. For example, one obtains $s \propto n^{1/(2a-1)}$ if $\max_{j=1}^{p} \|\mathrm{E}[X_j X]\|_2 \leq \text{const}$ or $s \propto n^{1/(2(a-1))}$ if $\max_{j=1}^{p} \|\mathrm{E}[X_j X]\|_\infty \leq \text{const}$ where $\propto$ means "is proportional to."

Therefore, if $s\log(\max\{p,n\})/n$ is small, Lasso and Post-Lasso regression come close to the population regression function/best linear predictor. Relative to our conjectured rate $\sqrt{s/n}$, there

is an additional factor $\sqrt{\log(\max\{p, n\})}$ in the bound. This factor captures the price of not knowing *a priori* which of the p regressors are the s important ones. Lasso approximately finds these important predictors, but correspondingly suffers a loss relative to a predictor estimated with knowledge of the best s-dimensional model ("oracle estimator"). A theoretical guarantee similar to Theorem 3.2.1 has been established for cross-validated Lasso [4], though with number of selected regressors diverging slowly relative to s rather than achieving $s = \text{const} \cdot s$.

Under approximate sparsity and with appropriate choice of penalty parameters, Lasso and Post-Lasso will approximate the best linear predictor well. Theoretically, they will not overfit the data, and we can thus use the sample and adjusted R^2 and MSE to assess out-of-sample predictive performance. Of course, it is always a good idea to verify the out-of-sample predictive performance by using sample splitting.

Remark 3.2.1 (Exact Sparsity) It is helpful to consider the exactly sparse case, in which there are only k non-zero coefficients bounded by some constant and the rest of the coefficients are exactly zero. In this case, the effective dimension is (up to constants) equal to the number of non-zero coefficients, i.e.

$$s = \text{const} \cdot k.$$

To see this, note that β satisfies the approximate sparsity condition with $A = \text{const} \cdot k^a$ for $a \geq 1$, since $\beta_j \leq \text{const} \leq \text{const} \cdot k^a / j^a$ for $j \leq k$ and $\beta_j = 0 \leq \text{const} \cdot k^a / j^a$ for $j > k$. Then $s \leq \text{const} \cdot k n^{1/2a}$, which yields the result as $a \to \infty$.

On regularity conditions*. A sufficient condition under which Theorem 3.2.1 can be established is the restricted isometry condition:

Definition 3.2.1 (Restricted Isometry) *The following conditions hold:*

$$\textit{Uniformly in } Z \subset X : \dim(Z) \leq L = s \log(n),$$

$$\sup_{\|a\|=1} |a'(\mathbb{E}_n[ZZ'] - \mathrm{E}[ZZ'])a| \approx 0,$$

$$0 < C_1 \leq \inf_{\|a\|=1} a'\mathrm{E}[ZZ']a \leq C_2 < \infty,$$

where C_1 and C_2 are constants.

This condition says that "small groups" of regressors are not

collinear and are well-behaved. I.e. we have that subvectors Z of X with dimension $L = s\log(n)$ have empirical Gram matrices $\mathbb{E}_n[ZZ']$ that are close to their population analogues $\mathrm{E}[ZZ']$ in the operator norm and have population covariance matrix $\mathrm{E}[ZZ']$ with eigenvalues bounded away from zero and from above. This condition is simple and intuitive but is stronger than necessary. Results similar to Theorem 3.2.1 have been shown to hold under considerably weaker conditions. The condition $\sup_{\|a\|=1} |a'(\mathbb{E}_n[ZZ'] - \mathrm{E}[ZZ'])a| \approx 0$ has been demonstrated to be valid under various more primitive conditions; see Appendix 3.C.

3.3 A Helicopter Tour of Other Penalized Regression Methods for Prediction

Instead of the Lasso penalty, other penalty schemes can be used, leading to different regression estimators with different properties. These estimators are motivated by different structures for the coefficients on the set of regressors in a high-dimensional model. We consider three important settings where coefficient are sparse, dense, or sparse+dense.

We have already seen that sparse coefficient vectors have a small number of relatively large, non-zero coefficients with the rest of the coefficients being close enough to zero to be ignorable. A dense coefficient vector has the vast majority or all coefficients non-zero and of comparable magnitude. A sparse+dense structure has the vast majority of coefficients being non-zero and of similar magnitude along with a small number of relatively large coefficients. Figure 3.5 illustrates each setting.

Throughout this section, we assume that regressors have been centered and normalized to have second empirical moment equal to 1. We thus ignore coefficient specific penalty parameters like the $\hat{\psi}_j$ in the Lasso problem (3.1.3).

We have already outlined Lasso regression, which performs best in an approximately sparse setting. We next consider the Ridge method, which performs best in the dense setting.

Ridge. The Ridge method estimates coefficients by penalized least squares, where we minimize the sum of squared prediction error plus the penalty term given by the sum of

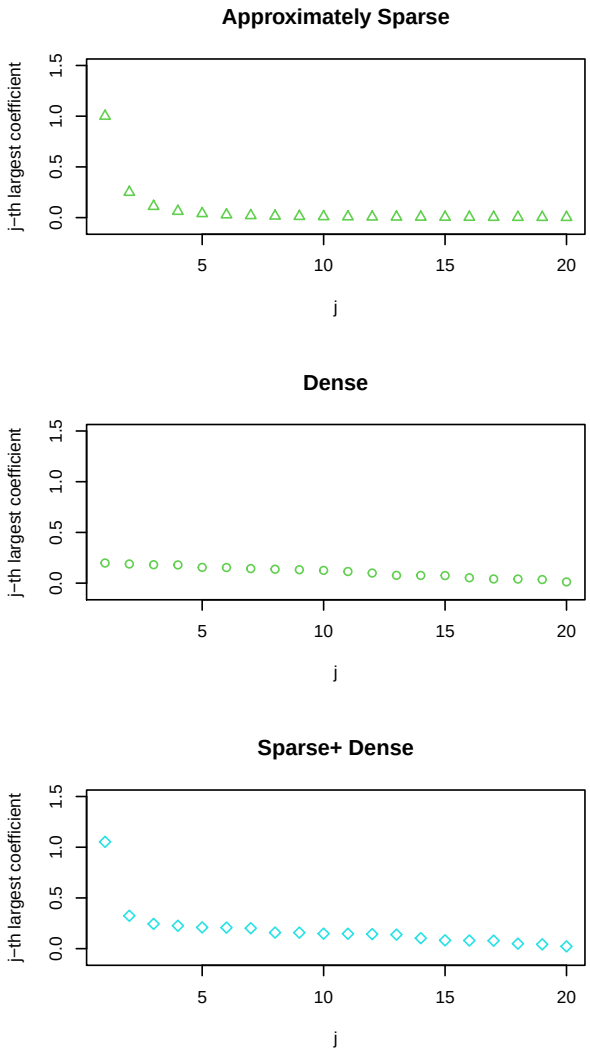

Figure 3.5: The Lasso penalty is best suited for approximately sparse models, and the Ridge penalty for models with small dense coefficients. The Elastic Net can be tuned to perform well with either sparse or dense coefficients. The Lava penalty is best suited for models with coefficients generated as the sum of approximately sparse coefficients and small dense coefficients.

the squared values of the coefficients times a penalty level λ:

$$\hat{\beta}(\lambda) = \arg\min_{b \in \mathbb{R}^p} \sum_{i=1}^{n} (Y_i - b'X_i)^2 + \lambda \sum_j b_j^2.$$

Ridge balances the complexity of the model measured by the sum of squared coefficients with the goodness of in-sample fit. In contrast to Lasso, Ridge penalizes the large values of coefficients much more aggressively and small values much less aggressively – indeed, squaring big values makes them even bigger and squaring small numbers makes them even smaller.

Because of the latter property,

- Ridge does not set estimated coefficients to zero and so it does not do variable selection.
- The Ridge predictor $\hat{\beta}'X$ is especially well suited for prediction in "dense" models, where the β_j's are all small without necessarily being approximately sparse.
- Ridge regression is also well suited when the matrix $\mathrm{E}[XX']$ is poorly behaved, as measured by the decay of its eigenvalues to zero.

In the dense case, the Ridge predictor can easily outperform the Lasso predictor.

Like Ridge, the Lasso predictor empirically seems to have reasonable prediction performance in the presence of ill-behaved design matrices, although we don't understand its theoretical properties well in this case.

Remark 3.3.1 (Theoretical Properties of the Ridge Procedure*) For excellent analysis of Ridge properties, see [7], who present the following bound for the fixed (conditional on) $X_1, \ldots X_n$ case holding with high probability:

$$\mathbb{E}_n\left[(\hat{\beta}'X - \beta'X)^2\right] \lesssim \sum_{j=1}^{p} \frac{\lambda^2 \zeta_j \gamma_j^2}{(\zeta_j^2 + \lambda)^2} + \frac{\mathrm{E}[\epsilon^2]}{n} \sum_{j=1}^{p} \left(\frac{\zeta_j^2}{(\zeta_j + \lambda)^2}\right),$$

where $(\zeta_j)_{j=1}^p$ are eigenvalues of $\mathbb{E}_n[XX']$ and γ_j are such that $\beta = \sum_{j=1}^p \gamma_j c_j$ with c_k being the eigenvectors of $\mathbb{E}_n[XX']$. The theoretically optimal penalty level can be chosen to minimize the right hand side, though doing so is infeasible as the right hand side depends on β. In practice, the penalty level is generally chosen by cross-validation. An analogous result holds for bounding $\mathrm{E}_X\left[(\hat{\beta}'X - \beta'X)^2\right]$ in the case of random $X_1, \ldots X_n$; see [7] for the statement.

The first component on the right hand side can be thought of as squared bias, and the second component is mean squared estimation error. Observe that when $\zeta_j = 1$ and λ is bounded, the second term is of order p/n, which translates to the rate of $\sqrt{p/n}$ after taking the square root. Having the second term go to 0 thus requires $\sqrt{p/n} \to 0$. In contrast, p can be larger than n and the second term can still vanish when eigenvalues

decay to zero. In this case, the effective dimension for a given λ is

$$d(\lambda) = \sum_{j=1}^{p} \frac{\zeta_j^2}{(\zeta_j + \lambda)^2},$$

and the second term is of order $d(\lambda)/n$. The ratio $d(\lambda)/n$ then determines the rate at which the Ridge predictor approximates the optimal predictor if the square bias term is of smaller order. Of course, it is hard to know that the square bias term is of smaller order than the second term in practice. The squared bias term will also not be of small order when there is a large γ_j associated with a large eigenvalue ζ_j.

Remark 3.3.2 (Connection to Principal Components*) Ridge regression is closely related to *principal components regression* which regresses an outcome of the first K principal components of the predictor variables X_i. Principal components provide mutually orthogonal rotations of the original X_i's that maximize fit to the overall design matrix. Here, we consider a case where we have $p < n$ centered predictor variables that are linearly independent. We let P_{ki} denote the i^{th} element of the k^{th} normalized principal component - the principal component divided by it's standard deviation which is given by the k^{th} largest eigenvalue of $\mathbb{E}_n[XX']$, ζ_k. Under these conditions, the ridge prediction can be expressed as

$$X_i'\hat{\beta} = \sum_{k=1}^{p} P_{ki} \frac{\zeta_k}{\zeta_k + \lambda} \mathbb{E}_n[P_k Y].$$

Note that principal components regression using the first K principal components would produce predictions

$$\hat{y}_i = \sum_{k=1}^{K} P_{ki} \mathbb{E}_n[P_k Y].$$

That is, Ridge and principal components regression are tightly connected. Unlike principal components regression, Ridge regression does not pre-select which principal components to use but instead places less weight on low variance principal components according to $\frac{\zeta_k}{\zeta_k+\lambda}$. We find the implicit use of principal components in ridge to be interesting, but note that we can explicitly use principal components as input variables in all penalized methods and in the more advanced methods that we discuss in Chapter 9. We visit using Principal Component Analysis for feature extraction when we outline feature engineering in Chapter 11. For further discussion, see

[8] p. 64-67 or the blog post Ridge vs PCA.

Ridge and Lasso have other useful modifications or hybrids that can perform well in the sparse, dense or sparse + dense settings. One popular modification is the Elastic Net [9] that can perform well in either the sparse or the dense scenario with appropriate tuning.

Elastic Net. The Elastic Net method estimates coefficients by penalized least squares with the penalty given by a linear combination of the Lasso and Ridge penalties:

$$\hat{\beta}(\lambda_1, \lambda_2) = \arg\min_{b \in \mathbb{R}^p} \sum_i (Y_i - b'X_i)^2 + \lambda_1 \sum_j b_j^2 + \lambda_2 \sum_j |b_j|.$$

We see that the penalty function has two penalty levels λ_1 and λ_2, which are chosen by cross-validation in practice.

- By selecting different values of penalty levels λ_1 and λ_2, we have more flexibility with Elastic Net for building a good prediction rule than with just Ridge or Lasso.
- The Elastic Net performs variable selection unless we completely shut down the Lasso penalty by setting $\lambda_2 = 0$.
- With proper tuning, Elastic Net works well in regression models where regression coefficients are either approximately sparse or dense.

See [10] for some theoretical results on Elastic Net.

Another way to combine the Lasso and Ridge penalties is the Lava method, which is intended to work well in sparse+dense settings.

Lava. The Lava method ([11], [12]) estimates coefficients by solving the penalized least squares problem:

$$\hat{\beta}(\lambda_1, \lambda_2) = \arg \min_{b: b = \delta + \xi \in \mathbb{R}^p} \sum_i (Y_i - b'X_i)^2$$

$$+\lambda_1 \sum_j \delta_j^2 + \lambda_2 \sum_j |\xi_j|.$$

Here components of the parameter vector are split into a "dense part" δ_j and "sparse part" ξ_j, where the δ_j's are penalized like in Ridge, and the ξ_j's are penalized like in Lasso. The minimization program automatically determines the best split into the dense and sparse parts. There are two corresponding penalty levels λ_1 and λ_2, which can be chosen by cross-validation in practice.

- Compared to the Elastic Net, the Lava method penalizes large and small coefficients much less aggressively – large coefficients are penalized like Lasso and small coefficients like Ridge. Like Ridge, Lava does not do variable selection.
- Lava is designed to work well in

  "sparse + dense"

  regression models where there are several large coefficients and many small coefficients that do not vanish quickly enough to satisfy approximately sparsity.
- With proper tuning that allows either λ_1 or λ_2 to be set to large values, Lava can also work in either "sparse" or "dense" models.

Theoretical guarantees for these methods are given in [11] and [12]. Theoretically and practically, Lava can significantly outperform Lasso, Ridge and Elastic Net in "sparse+dense" models, and, with appropriate tuning, has comparable performance to Lasso in "sparse" models and to Ridge in "dense" models.

3.4 Choice of Regression Methods in Practice

How should we select the appropriate penalized regression method? The answer is simple if we are interested in building the best prediction. We can split the data into training and testing sets and simply choose the method that performs the best on the test set. Rigorous theoretical guarantees for this approach have been provided by [13].

We show an example of this approach in R Notebook on ML for Prediction of Wages and Python Notebook on ML for Prediction of Wages which illustrate the use of penalized regression methods for predicting log-wages using CPS 2015 data. We can

also use ensemble methods to aggregate prediction methods to get boosts in predictive performance – we describe these aggregation methods in Chapter 9.

Notebooks

- R Notebook on Penalized Regressions and Python Notebook on Penalized Regressions provide details of implementation of different penalized regression methods and examine their performance for approximating regression functions in a simulation experiment. The simulation experiment includes one case with approximate sparsity, one case with dense coefficients, and another case with both approximately sparse and dense components.

- R Notebook on ML for Prediction of Wages and Python Notebook on ML for Prediction of Wages provide details of implementation of different penalized regression methods and examine their performance for predicting log-wages using CPS 2015 data.

Notes

Lasso was introduced by Frank and Friedman [14], and its geometric and computational properties were elaborated on by Tibshirani [15], who also gave it its name. The first general theoretical analysis of Lasso was done by Bickel, Ritov, and Tsybakov [2]. Hastie, Tibshirani, and Wainwright [16] provides a good textbook introduction.

There are many variations on the basic Lasso theme, only some of which we mentioned in this chapter. The properties of the Post-Lasso estimator in approximately sparse models (without assuming that Lasso perfectly selects the "right model") were first established in [3]. The properties of Lasso and Post-Lasso don't hinge on the assumption of Gaussian or sub-Gaussian errors, as proven in [6], though such assumptions are often imposed. Fundamentally, the properties of these procedures rely on a high-dimensional central limit theorem ([5]) that allows Gaussian approximations to key average-like quantities.While cross-validation has been frequently used to select the penalty level, validity of this approach for Lasso was only proven recently – [4]. The Lasso has been extended to clustered dependence by [17] and to time series and many time series by [18], with the corresponding package available at this Link.

There is a large literature on Ridge estimation, with the reference [7] providing what seems to be the state of the art. The Lava approach has been proposed and analyzed in [11] and [12]. [12] also discusses applications to problems with latent confounding and, for this reason, refers to Lava as the spectral deconfounder. We discuss other approaches to dealing with latent confounding in Chapter 12 and Chapter 13.

Study Problems

1. Solve the Lasso optimization problem analytically with only one regressor and interpret the solution.

2. Experiment with the R Notebook on Penalized Regressions, trying out modifications of the Monte-Carlo experiments. As examples, you might change parameters that govern the speed of decay of coefficients to zero, change the error distribution, or alter the structure of dependence among the design variables. Try to explain the results to a fellow student, linking explanations to the theoretical properties of these methods.

3. Experiment with the R Notebook on ML Prediction of Wages. Try to explain the results to a fellow student, linking explanations to the theoretical properties of these methods.

3.A Additional Discussion and Results

Iterative Estimation of σ

The plug-in choice of λ given in equation (3.1.4) requires an estimate of σ. We can estimate σ using the following iterative method. Let X^0 be a small set of regressors (a trivial choice is just the intercept, but we may include, for example, the five regressors that are most strongly correlated with the Y_i's). Let $\widehat{\beta}_0$ be the least squares estimator of the coefficients on the covariates associated with X^0, and define

$$\hat{\sigma}_0 := \sqrt{\mathbb{E}_n[(Y_i - \hat{\beta}_0' X_i^0)^2]}.$$

Set $k = 0$, and specify a small constant $\nu \geq 0$ as a tolerance level and a constant $K > 1$ as an upper bound on the number of iterations:

We find that $K = 1$ works well in practice.

1. Compute the Lasso estimator $\widehat{\beta}$ based on the penalty level λ given in equation (3.1.4) using $\hat{\sigma}_k$.
2. Set $\hat{\sigma}_{k+1} = \sqrt{\mathbb{E}_n[(Y_i - \hat{\beta}' X_i)^2]}$.
3. If $|\hat{\sigma}_{k+1} - \hat{\sigma}_k| \leqslant \nu$ or $k > K$, stop; otherwise set $k \leftarrow k + 1$ and go to (1).

We note that the plug-in choice of λ given in equation (3.1.4) relies on assuming homoskedasticity of the BLP residuals, i.e. $\epsilon \perp\!\!\!\perp X$. This independence implies that $\mathrm{E}[\epsilon^2 X_j^2] = \mathrm{E}[\epsilon^2]\mathrm{E}[X_j^2]$. With independent observations where we do not have $\epsilon \perp\!\!\!\perp X$, we should use penalty loadings $\hat{\psi}_j = \sqrt{\mathbb{E}_n[\hat{\epsilon}^2 X_j^2]}$, where $\hat{\epsilon}_i \approx \epsilon_i$ can be estimated in a similar iterative manner as described above. In this case, we would then take $\hat{\sigma} = 1$ in formula (3.1.4) for λ (see [6] for more details).

We expect the homoskedastic formula for the penalty provided in (3.1.4) will work well in many cases, especially when random variables ϵ, X_j are expected to have fast decaying tail probabilities. For example, when fourth moments of ϵ, X_j are bounded by some constant factor of their second moments, an application of the Cauchy-Schwarz inequality implies that $\mathrm{E}[\epsilon^2 X_j^2] \leq \text{const} \cdot \mathrm{E}[\epsilon^2]\mathrm{E}[X_j^2]$, which is, up to a constant, the simplifying condition implied by homoskedasticity.

Some Lasso Heuristics via Convex Geometry*

Assume $\hat{\psi}_j = 1$ for each j for simplicity, which amounts to normalizing regressors to have the second empirical moment equal to 1. Consider

$$\widehat{\beta} \in \arg\min_{b \in \mathbb{R}^p} \widehat{Q}(b) + \frac{\lambda}{n}\|b\|_1, \tag{3.A.1}$$

where

$$\widehat{Q}(b) = \mathbb{E}_n[(Y_i - b' X_i)^2].$$

The key quantity in the analysis of (3.A.1) is the score – the gradient of $\widehat{Q}$ at the true value:

$$S = -\nabla \widehat{Q}(\beta_0) = 2\mathbb{E}_n[X\epsilon].$$

The score S is the effective "noise" in the problem that should be dominated by the regularization. However, we would like to

make the regularization bias as small as possible. This reasoning suggests choosing the smallest penalty level λ that is just large enough to dominate the noise with high probability, say $1 - \mathsf{a}$, which yields

$$\lambda > c\Lambda, \text{ for } \Lambda := n\|S\|_\infty. \tag{3.A.2}$$

Here, Λ is the maximal score scaled by n, and $c > 1$ is a theoretical constant that guarantees that the score is dominated.

It is useful to mention some simple heuristics for the principle (3.A.2) which arise from considering the simplest case where all of the regressors are irrelevant so that $\beta = 0$. We want our estimator to perform at a near-oracle level in all cases, including this case, but here the oracle estimator β^* sets $\beta^* = \beta = 0$. We thus also want $\widehat{\beta} = \beta = 0$ in this case, at least with a high probability, say $1 - \mathsf{a}$. From the subgradient optimality conditions for (3.A.1), we must have

$$-S_j + \lambda/n > 0 \text{ and } S_j + \lambda/n > 0 \text{ for all } 1 \leq j \leq p \tag{3.A.3}$$

for the Lasso estimator for each coefficient to be exactly 0. We can guarantee (3.A.3) holds by setting the penalty level λ/n such that $\lambda > n \max_{1\leq j\leq p} |S_j| = n\|S\|_\infty$ with probability at least $1 - \mathsf{a}$, which is precisely what the rule (3.A.2) does.

Gaussian approximations to this score motivate the following X-dependent penalty implementation.

Remark 3.A.1 (Refining Penalty Levels) An X-dependent penalty level can be specified as follows:

$$\lambda = c \cdot 2\hat{\sigma}\Lambda(1 - \mathsf{a}|\{X_i\}_{i=1}^n), \tag{3.A.4}$$

where

$$\Lambda(1 - \mathsf{a}|\{X_i\}_{i=1}^n) \\ = (1 - \mathsf{a}) - \text{quantile of } \; n\|\mathbb{E}_n[Xg/\Psi]\|_\infty \mid \{X_i\}_{i=1}^n,$$

g_i are i.i.d. $N(0, 1)$, and $\Psi = \text{diag}(\hat{\psi}_j)_{j=1}^p$. $\Lambda(1 - \mathsf{a}|\{X_i\}_{i=1}^n)$ can be thus be easily approximated by simulation. The use of normal errors g_i could be motivated by assuming the Gaussian errors ϵ_i in the model or by appealing to a high-dimensional central limit theorem. We note that by the union

bound and Feller's tail inequality,

$$\begin{aligned}\Lambda(1-\mathsf{a}|\{X_i\}_{i=1}^n) &\leq \sqrt{n}z_{1-\mathsf{a}/(2p)} \\ &\leq \sqrt{2n\log(2p/\mathsf{a})}.\end{aligned} \tag{3.A.5}$$

Thus, $\sqrt{2n\log(2p/\mathsf{a})}$ provides a simple upper bound on the penalty level.

Refined penalty levels are important when components of X_i are highly correlated, in which case the X-dependent penalty will be much lower that the bounds given in 3.A.5. Using the lower penalty level can offer both practical and theoretical boosts in performance in such cases.

Other Variations on Lasso

Here and below we assume that

$$\hat{\psi}_j = 1, \quad j = 1, \ldots, p$$

to simplify notation. A variant of Lasso, called the *Square-root Lasso* estimator ([19],[20]), is defined as a solution to the following program:

$$\min_{b\in\mathbb{R}^p} \sqrt{\mathbb{E}_n[(Y-b'X)^2]} + \frac{\lambda}{n}\|b\|_1. \tag{3.A.6}$$

Analogously to Lasso, we may set the penalty level as

$$\lambda = c\cdot\widetilde{\Lambda}(1-\mathsf{a}|\{X_i\}_{i=1}^n), \tag{3.A.7}$$

where $c > 1$ and

$$\begin{aligned}&\widetilde{\Lambda}(1-\mathsf{a}|\{X_i\}_{i=1}^n) \\ &\quad = (1-\mathsf{a}) - \text{quantile of } \; n\|\mathbb{E}_n[Xg]\|_\infty/\sqrt{\mathbb{E}_n[g^2]} \mid \{X_i\}_{i=1}^n,\end{aligned}$$

with $g_i \sim N(0,1)$ independent for $i = 1, \ldots, n$. As with Lasso, there is also a simple asymptotic option for setting the penalty level:

$$\lambda = c\cdot 2\sqrt{n}z_{1-\mathsf{a}/(2p)}. \tag{3.A.8}$$

The main attractive feature of (3.A.6) is that the penalty level λ specified above is independent of the value σ. This estimator has statistical performance that is as good as the iterative or cross-validated Lasso. Moreover, the estimator is a solution to a

highly tractable conic programming problem:

$$\min_{t\geq 0, b\in\mathbb{R}^p} t + \frac{\lambda}{n}\|b\|_1 : \sqrt{\mathbb{E}_n[(Y - b'X)^2]} \leq t, \tag{3.A.9}$$

where the criterion function is linear in parameters t and positive and negative components of b, while the constraint can be formulated with a second-order cone, informally known as the "ice-cream cone."

There are several other estimators that make use of penalization by the ℓ_1-norm. A final important case is the *Dantzig selector* estimator [21]. It also relies on ℓ_1-regularization but exploits the notion that the residuals should be nearly uncorrelated with the covariates. The estimator is defined as a solution to

$$\min_{b\in\mathbb{R}^p} \|b\|_1 \ : \ \|\mathbb{E}_n[X(Y - b'X)]\|_\infty \leq \lambda/n. \tag{3.A.10}$$

Again, one may set $\lambda = \sigma\Lambda(1-\mathsf{a}|\{X_i\}_{i=1}^n)$. Here, we focused our discussion on Lasso but virtually all theoretical results carry over to other ℓ_1-regularized estimators including (3.A.6) and (3.A.10). We also refer to [22] for a feasible Dantzig estimator that combines the square-root Lasso method (3.A.9) with the Dantzig method.

3.B Cross-Validation

Cross-validation is a common practical tool that provides a way to choose tuning parameters such as the penalty level in Lasso. The idea of cross-validation is to rely on repeated splitting of the training data to estimate the out-of-sample predictive performance.

Definition 3.B.1 (Cross-Validation in Words)

- *We partition the data into K blocks called "folds." For example, with $K = 5$, we split the data into 5 non-overlapping blocks.*
- *Leave one block out. Fit a prediction rule on all the other blocks. Predict the outcome observations in the left out block, and record the empirical Mean Squared Prediction Error. Repeat this for each block.*
- *Average the empirical Mean Squared Prediction Errors over blocks.*
- *We do these steps for several or many values of the tuning*

parameters and choose the value of the tuning parameter that minimizes the Averaged Mean Squared Prediction Error.

We can also consider many different methods for constructing prediction rules as well. For example, we could try Lasso with many different values of the penalty parameter and Ridge with many different values of the penalty parameter and choose the tuning parameter and method (Lasso or Ridge) that minimizes the cross-validated Mean Squared Prediction Error.

Definition 3.B.2 (Cross-Validation: Formal Description)

- *Randomly partition the observation indices* $1, ..., n$ *into* K *folds* $B_1, ..., B_K$.
- *For each* $k = 1, ..., K$, *fit a prediction rule denoted by* $\hat{f}^{[k]}(\cdot; \theta)$, *where* θ *denotes the tuning parameters such as penalty levels and* $\hat{f}^{[k]}$ *depends only on observations with indices not in the fold* B_k.
- *For each* $k = 1, ..., K$, *the empirical out-of-sample MSE for the block* B_k *is*

$$MSE_k(\theta) = \frac{1}{m_k} \sum_{i \in B_k} (Y_i - \hat{f}^{[k]}(X_i; \theta))^2,$$

  where m_k *is the size of the block* B_k.
- *Compute the cross-validated MSE as*

$$CV\text{-}MSE(\theta) = \frac{1}{K} \sum_{k=1}^{K} MSE_k(\theta).$$

- *Choose the tuning parameter* $\hat{\theta}$ *as a minimizer of* $CV\text{-}MSE(\theta)$.

Remark 3.B.1 (On Guarantees of Cross-Validated Predictors) A common step people do in practice is to retrain the predictor $\hat{f}(X)$ on the entire data with the best tuning parameter $\hat{\theta}$ found by cross-validation. Theoretical properties of the resulting cross-validated predictor $\hat{f}(X)$ are only well understood for some high-dimensional problems. E.g., see [4] for results on Lasso with cross-validation.

Remark 3.B.2 (Guarantees for Pooled Cross-Validated Estimator) On the other hand, there are rigorous theoretical guarantees for the pooled cross-validated predictor:

$$\hat{f}(X) = \frac{1}{K} \sum_{k=1}^{K} \hat{f}^{[k]}(X; \hat{\theta}),$$

which are provided by [23] and [13] who establish that the resulting prediction rule has optimal or near-optimal rates for approximating the best predictor in a given class.

Note that the pooled procedure is different from the default CV procedure implemented in many software packages and used in many applications.

3.C Laws of Large Numbers for Large Matrices*

The following results are useful for justifying the restricted isometry condition for empirical Gram matrices $\mathbb{E}_n[XX']$.

Let s_n, p_n, k_n be sequences of positive constants, $\ell_n = \log(n)$, and C a fixed positive constant. Let $(X_i)_{i=1}^n$ be iid. vectors. Denote by $(Z_i)_{i=1}^n$ corresponding subvectors.

Suppose that $\max_{\|a\|=1} \mathrm{E}[(Z'a)^2] \leq C$ for all $Z \subset X$ such that $\dim(Z) \leq s_n \ell_n$ and that one of the following holds:

(a) X_i is a sub-Gaussian, namely

$$\sup_{\|u\|\leq 1} \mathrm{P}(|X_i'u| > t) \leq 2\exp(-t^2/c_2^2)$$

for all $t \geq 0$, and $s_n(\log n)\left(\log\left(\max\{p_n, n\}\right)\right)/n \to 0$;

(b) X_i has bounded components, namely

$$\max_j |X_{ij}| \leq k_n$$

and $k_n^2 s_n \log^2 n \log(s_n \log n) \log\left(\max\{p_n, n\}\right)/n \to 0$.

Then with probability $1 - \delta_n$

$$\max_{Z\subset X:\dim(Z)\leq s_n\ell_n} \max_{\|a\|=1} |a'\left(\mathbb{E}_n\left[ZZ'\right] - \mathrm{E}\left[ZZ'\right)\right]a| \leq \Delta_n,$$

where (δ_n, Δ_n) are decreasing sequences and $(\delta_n, \Delta_n) \to 0$.

Under (a) the result follows from Theorem 3.2 in [24] and under (b) the result follows from [25]. These references also imply finite-sample characterization of error bounds (δ_n, Δ_n).

3.D A Sketch of the Lasso Guarantee Under Exact Sparsity*

Let us assume that the population BLP β_0 satisfies exact sparsity, i.e. only s out of p coefficients are non-zero. Denote with A the set of non-zero coefficients and with A^c the complement of that set. Since the Lasso minimizes the objective $\hat{Q}(b) + \frac{\lambda}{n}\|b\|_1$ for $\widehat{Q}(b) = \mathbb{E}_n[(Y - b'X)^2]$, we have

$$\hat{Q}(\hat{\beta}) - \hat{Q}(\beta_0) \leq \frac{\lambda}{n}(\|\beta_0\|_1 - \|\hat{\beta}\|_1). \tag{3.D.1}$$

Let $\nu := \hat{\beta} - \beta_0$. Since the objective $\hat{Q}(\beta)$ is convex in β, we have by an application of the Cauchy-Schwarz inequality that

$$\hat{Q}(\hat{\beta}) - \hat{Q}(\beta_0) \geq \nabla\hat{Q}(\beta_0)'\nu = -S'\nu \geq -\|S\|_\infty\|\nu\|_1$$

for $S = -\nabla\widehat{Q}(\beta_0) = 2\mathbb{E}_n[X\epsilon]$.

We will assume that λ is chosen such that we have $\frac{\lambda}{n} \geq 2\|S\|_\infty$ with probability $1 - \mathsf{a}$.[12] We focus then on the good event where the above inequality is satisfied. Then we can combine the above two inequalities:

$$\frac{\lambda}{n}(\|\beta_0\|_1 - \|\hat{\beta}\|_1) \geq -\|S\|_\infty\|\nu\|_1 \geq -\frac{\lambda}{2n}\|\nu\|_1.$$

Hence with with high probability,

$$\hat{\beta} - \beta_0 \in RC = \{\nu : \|\beta_0 + \nu\|_1 \leq \|\beta_0\|_1 + \|\nu\|_1/2\}.$$

Note also that $\nu \in RC$ implies[13]

$$\|\nu_{A^c}\|_1 \leq 3\|\nu_A\|_1 \tag{3.D.2}$$

where ν_A denotes the entries from ν in A and ν_{A^c} denotes the entries of ν in A^c. This inequality roughly states that the error vector $\nu = \hat{\beta} - \beta_0$ is primarily supported on A.

We impose the following regularity condition:

$$0 < C_1 \leq \min_{\nu \in RC \setminus 0} \frac{\nu'\mathrm{E}[XX']\nu}{\|\nu\|^2} \leq C_2 < \infty. \tag{3.D.3}$$

The restricted isometry conditions we impose in the text are known to imply this condition.[14]

Suppose that we can argue that we have, for any vector $\nu \in$

12: The High-Dimensional CLT bounds tell us that if we set $\lambda \approx \sqrt{n\log(\max\{p/\mathsf{a}, n\})}$, then this inequality holds with probability $1 - \mathsf{a}$.

13: Verify this as a reading exercise.

14: See, e.g. Lemma 10 in [26] for an argument based on [2].

RC,

$$\nu' \mathbb{E}_n[XX']\nu \geq \hat{C}_1 \|\nu\|_2^2 \tag{3.D.4}$$

for some $\hat{C}_1 > 0$ that will be generally be related to C_1 and features of the population. (3.D.4) is oftentimes referred to as the empirical Restricted Strong Convexity (RSC) property. We provide an example and the corresponding $\hat{C}_1$ below.

Then, using the fact that $\hat{Q}(\beta)$ is quadratic in β, we can invoke the exact second order Taylor expansion:

$$\hat{Q}(\hat{\beta}) - \hat{Q}(\beta_0) = S'\nu + \nu' \mathbb{E}_n[XX']\nu \geq -\|S\|_\infty \|\nu\|_1 + \hat{C}_1 \|\nu\|_2^2.$$

When combined with the upper bound from the optimality of $\hat{\beta}$ for the penalized empirical loss and the fact that $\frac{\lambda}{n} \geq 2\|S\|_\infty$, this expansion yields

$$\frac{\lambda}{n}\|\nu\|_1 \geq \hat{Q}(\hat{\beta}) - \hat{Q}(\beta_0) \geq -\frac{\lambda}{2n}\|\nu\|_1 + \hat{C}_1 \|\nu\|_2^2.$$

The second crucial inequality that

$$\|\nu\|_2^2 \leq \frac{3\lambda}{2\hat{C}_1 n}\|\nu\|_1 \tag{3.D.5}$$

then follows.

Finally, note that for any vector ν that is primarily supported on A, the ℓ_2 and ℓ_1 norms are within a factor $\approx \sqrt{s}$ of each other:

$$\|\nu\|_1 = \|\nu_A\|_1 + \|\nu_{A^c}\|_1 \leq 4\|\nu_A\|_1 \leq 4\sqrt{s}\|\nu_A\|_2 \leq 4\sqrt{s}\|\nu\|_2$$

where we used the norm inequality, that for an s-dimensional vector v, we have $\|v\|_1 \leq \sqrt{s}\|v\|_2$. Thus, we can conclude

$$\|\nu\|_2 \leq \frac{6\lambda}{\hat{C}_1 n}\sqrt{s}. \tag{3.D.6}$$

Using the assumption that $\nu' \mathrm{E}[XX']\nu \leq C_2\|\nu\|_2$ for $\nu \in RC$, we get the final bound:

$$\sqrt{\mathrm{E}_X[(X'\hat{\beta} - X'\beta_0)^2]} = \sqrt{\nu' \mathrm{E}[XX']\nu} \leq C_2\|\nu\|_2 \leq \frac{6\lambda\, C_2}{\hat{C}_1 n}\sqrt{s}.$$

It remains to argue the empirical RSC property. Note that if

$$\|\mathbb{E}_n[XX'] - \mathrm{E}[XX']\|_\infty \leq \mu_n$$

with probability approaching 1,[15] then we have

15: Application of the high-dimensional CLT implies that we can take $\mu_n \propto \sqrt{\frac{\log(\max\{p,n\})}{n}}$.

$$\begin{aligned} v'\mathbb{E}_n[XX']v &\geq v'\mathrm{E}[XX']v - \|v\|_1^2 \|\mathbb{E}_n[XX'] - \mathrm{E}[XX']\|_\infty \\ &\geq (C_1 - 16s\mu_n)\|v\|_2^2 \end{aligned}$$

by Condition (3.D.3) and an application of the Hölder inequality. Thus, if n is large enough such that $16s\mu_n \leq \frac{C_1}{2}$, we conclude that the empirical RSC condition holds with $\hat{C}_1 = \frac{C_1}{2}$.

Bibliography

[1] Antoine de Saint-Exupéry. *Terre des hommes*. Gallimard, 1939 (cited on page 69).

[2] Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. 'Simultaneous analysis of Lasso and Dantzig selector'. In: *Annals of Statistics* 37.4 (2009), pp. 1705–1732 (cited on pages 76, 87, 95).

[3] Alexandre Belloni and Victor Chernozhukov. 'Least Squares After Model Selection in High-dimensional Sparse Models'. In: *Bernoulli* 19.2 (2013). ArXiv, 2009, pp. 521–547 (cited on pages 76–79, 87).

[4] Denis Chetverikov, Zhipeng Liao, and Victor Chernozhukov. 'On cross-validated lasso in high dimensions'. In: *Annals of Statistics* 49.3 (2021), pp. 1300–1317 (cited on pages 76, 80, 87, 93).

[5] Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. 'Central Limit Theorems and Bootstrap in High Dimensions'. In: *Annals of Probability* 45.4 (2017), pp. 2309–2352 (cited on pages 76, 87).

[6] Alexandre Belloni, Daniel L. Chen, Victor Chernozhukov, and Christian B. Hansen. 'Sparse Models and Methods for Optimal Instruments with an Application to Eminent Domain'. In: *Econometrica* 80.6 (2012). Arxiv, 2010, pp. 2369–2429 (cited on pages 79, 87, 89).

[7] Daniel Hsu, Sham M. Kakade, and Tong Zhang. 'Random design analysis of ridge regression'. In: *Conference on Learning Theory*. Vol. 23. JMLR Workshop and Conference Proceedings. 2012, pp. 9.1–9.24 (cited on pages 83, 88).

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics New York, 2001 (cited on page 85).

[9] Hui Zou and Trevor Hastie. 'Regularization and variable selection via the elastic net'. In: *Journal of the Royal Statistical Society: Series B* 67.2 (2005), pp. 301–320 (cited on page 85).

[10] Christine De Mol, Ernesto De Vito, and Lorenzo Rosasco. 'Elastic-net regularization in learning theory'. In: *Journal of Complexity* 25.2 (2009), pp. 201–230. DOI: https://doi.org/10.1016/j.jco.2009.01.002 (cited on page 85).

[11] Victor Chernozhukov, Christian Hansen, and Yuan Liao. 'A lava attack on the recovery of sums of dense and sparse signals'. In: *Annals of Statistics* 45.1 (2017), pp. 39–76 (cited on pages 85, 86, 88).

[12] Domagoj Ćevid, Peter Bühlmann, and Nicolai Meinshausen. 'Spectral deconfounding via perturbed sparse linear models'. In: *Journal of Machine Learning Research* 21 (2020), pp. 1–41 (cited on pages 85, 86, 88).

[13] Marten Wegkamp. 'Model selection in nonparametric regression'. In: *Annals of Statistics* 31.1 (2003), pp. 252–273 (cited on pages 86, 94).

[14] lldiko E. Frank and Jerome H. Friedman. 'A statistical view of some chemometrics regression tools'. In: *Technometrics* 35.2 (1993), pp. 109–135 (cited on page 87).

[15] Robert Tibshirani. 'Regression shrinkage and selection via the Lasso'. In: *Journal of the Royal Statistical Society: Series B* 58.1 (1996), pp. 267–288 (cited on page 87).

[16] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015 (cited on page 87).

[17] Alexandre Belloni, Victor Chernozhukov, Christian Hansen, and Damian Kozbur. 'Inference in High-Dimensional Panel Models With an Application to Gun Control'. In: *Journal of Business & Economic Statistics* 34.4 (2016), pp. 590–605 (cited on page 87).

[18] Victor Chernozhukov, Wolfgang Karl Härdle, Chen Huang, and Weining Wang. 'Lasso-driven inference in time and space'. In: *Annals of Statistics* 49.3 (2021), pp. 1702–1735 (cited on page 87).

[19] Alexandre Belloni, Victor Chernozhukov, and Lie Wang. 'Square-root lasso: pivotal recovery of sparse signals via conic programming'. In: *Biometrika* 98.4 (2011). Arxiv, 2010, pp. 791–806 (cited on page 91).

[20] Alexandre Belloni, Victor Chernozhukov, and Lie Wang. 'Pivotal estimation via square-root lasso in nonparametric regression'. In: *Annals of Statistics* 42.2 (2014), pp. 757–788 (cited on page 91).

[21] Emmanuel Candès and Terence Tao. 'The Dantzig selector: statistical estimation when p is much larger than n'. In: *Annals of Statistics* 35.6 (2007), pp. 2313–2351 (cited on page 92).

[22] Eric Gautier and Alexander B. Tsybakov. 'High-Dimensional Instrumental Variables Regression and Confidence Sets'. In: *ArXiv working report* (2011) (cited on page 92).

[23] Guillaume Lecué and Charles Mitchell. 'Oracle inequalities for cross-validation type procedures'. In: *Electronic Journal of Statistics* 6 (2012), pp. 1803–1837 (cited on page 94).

[24] M. Rudelson and S. Zhou. 'Reconstruction from anisotropic random measurements'. In: *ArXiv:1106.1151* (2011) (cited on page 94).

[25] Mark Rudelson and Roman Vershynin. 'On sparse reconstruction from Fourier and Gaussian measurements'. In: *Communications on Pure and Applied Mathematics* 61.8 (2008) (cited on page 94).

[26] Alexandre Belloni and Victor Chernozhukov. 'High Dimensional Sparse Econometric Models: An Introduction'. In: *Inverse Problems and High-Dimensional Estimation: Stats in the Château Summer School, August 31 - September 4, 2009*. Ed. by Pierre Alquier, Eric Gautier, and Gilles Stoltz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 121–156. DOI: 10.1007/978-3-642-19989-9_3 (cited on page 95).