Predictive Inference 2: Modern High Dimensional Linear Regression

Alexander Quispe[1]

[1]PUCP

September 10, 2021

Analysis of Variance (ANOVA)

POPULATION	SAMPLE
$$\begin{aligned} Y &= \beta' X + \varepsilon, \quad E\varepsilon X = 0 \\ EY^2 &= E(\beta' X)^2 + E\varepsilon^2 \\ MSE_{pop} &= E\varepsilon^2 \\ R^2_{pop} &:= \frac{E(\beta' X)^2}{EY^2} = \\ & 1 - \frac{E\varepsilon^2}{EY^2} \quad \in \quad [0, 1] \end{aligned} \tag{1}$$	$$\begin{aligned} Y_i &= \hat{\beta}' X_i + \hat{\varepsilon}_i \\ \mathbb{E}_n Y_i^2 &= \mathbb{E}_n(\hat{\beta}' X_i)^2 + \mathbb{E}_n \hat{\varepsilon}_i^2 \\ MSE_{sample} &= \mathbb{E}_n \hat{\varepsilon}_i^2 \\ R^2_{sample} &:= \frac{\mathbb{E}_n(\hat{\beta}' X_i)^2}{\mathbb{E}_n Y_i^2} = \\ & 1 - \frac{\mathbb{E}_n \hat{\varepsilon}_i^2}{\mathbb{E}_n Y_i^2} \quad \in \quad [0, 1] \end{aligned} \tag{2}$$

By law of large numbers when p/n is small and n is large:

$$\begin{gathered} \mathbb{E}_n Y_i^2 \approx EY^2, \ \ \mathbb{E}_n(\hat{\beta}' X_i)^2 \approx E(\beta' X)^2, \ \ \mathbb{E}_n \hat{\varepsilon}_i^2 \approx E\varepsilon^2 \\ R^2_{sample} \approx R^2_{pop} \ \ and \ \ MSE_{sample} \approx MSE_{pop} \end{gathered} \tag{3}$$

Overfitting: What happens when p/n is not small

When p/n is not small, the discrepancy between the in-sample and out-of-sample measures of fit can be substantial. Let´s check the next example :

- $$\begin{aligned} X \sim N(0, I_p) \ \ and \ \ Y \sim N(0,1), \ \ \beta' X = 0, \ \ R^2_{pop} = 0 \\ if \ \ p = n, \ \ then \ \ R^2_{sample} \ \ is \ \ 1 \gg 0 \\ if \ \ p = \frac{n}{2}, \ \ then \ \ R^2_{sample} \ \ is \ \ about \ \ 0.5 \gg 0 \\ if \ \ p = \frac{n}{20}, \ \ then \ \ R^2_{sample} \ \ is \ \ about \ \ 0.05 \end{aligned} \tag{4}$$

  Better measures of out-of-sample predictive ability are the “adjusted" R^2 and MSE.

$$MSE_{adjusted} = \frac{n}{n-p} \mathbb{E}_n \hat{\varepsilon}_i^2, \ \ R^2_{adjusted} := 1 - \frac{n}{n-p} \frac{\mathbb{E}_n \hat{\varepsilon}_i^2}{\mathbb{E}_n Y_i^2} \tag{5}$$

Measuring Predictive Ability by Sample Splitting

To measure out-of-sample performance: **Data splitting.** The idea can be summarized in two parts:

1. Use a random part of data, called the **training sample**, for estimating/training the prediction rule.
2. Use the other part, called the **testing sample**, to evaluate the quality of the prediction rule, recording out-of-sample mean squared error and R^2.

Generic Evaluation of Prediction Rules by Sample-Splitting

1. Randomly partition the data into training and testing samples. Suppose we use n observations for training and m for testing/validation.
2. Use the training sample to compute a prediction rule $\hat{f}(X)$, for example, $\hat{f}(X) = \beta' X$.
3. Let V denote the indexes of the observations in the test sample. Then the out-of-sample/test mean squared error is

$$MES_{test} = \frac{1}{m} \sum_{k \in V} (Y_k - \hat{f}(X_k))^2 \tag{6}$$

and the out-of-sample/test R^2 is

$$R^2_{test} = 1 - \frac{MSE_{test}}{\frac{1}{m} \sum_{k \in V} Y_k^2} \tag{7}$$

Regression in a High-Dimensional Setting / LASSO

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

$$\min_{b \in \mathbb{R}^p} \sum_i (Y_i - b' X_i)^2 + \lambda \cdot \sum_{j=1}^{p} |b_j| \tag{8}$$

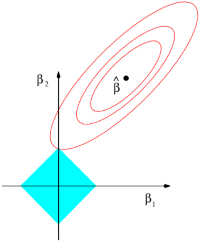
1. The first term is n times the sample mean square error
2. The second term is a penalty term, which penalizes the size of coefficients b_j by their absolute values times the penalty level λ A crucial point is the choice of the penalization parameter λ.
3. A theoretically valid choice is (Belloni Chernozhukov, 2013)

$$\lambda = 2.c\widehat{\sigma}\sqrt{2nlog(2p/\gamma)}, \;\; \widehat{\sigma} \approx \sigma = \sqrt{E \in^2} \tag{9}$$

4. Another good way to pick penalty level is by cross-validation (Chetverikov et al, 2020)

Contours of the error and constraint functions for the lasso

Figure 1: Lasso optimization with two coefficients.



Intuition : 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:

$$\widehat{\beta}_j = 0 \;\; if \;\; \left| \partial_{b_j} \sum_i (Y_i - \widehat{\beta}' X_i)^2 \right| < \lambda \tag{10}$$

OLS post Lasso

We can then **use the Lasso-selected set of regressors** to refit the model by least squares. This method is called the “least squares post Lasso" or simply **post-Lasso**

$$\widetilde{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \sum_i (Y_i - X_i'\beta)^2 : \beta_j = 0 \;\; if \;\; \widehat{\beta}_j = 0 \;\; for \;\; each \;\; j \tag{11}$$

Under approximate sparsity Lasso and Post-Lasso will approximate the best linear predictor well. This means that they won’t overfit the data, and we can 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

How to select λ

Big lambdas tend to result in a lot of shrinkage and sparsity, as $\lambda --> 0$ our solution approaches the OLS solution

Two ways to select λ

- Select model with lowest AIC/BIC/other plug-in criterion. This uses no out-of-sample information for selection but is fast.
- Cross-validate by testing on our hold-out test sample. **Variants of cross-validation are most commonly used.**

k-fold cross-validation

The next algorithm is taken from Ivan Rudik' Lectures Note in Dynamic Optimization(Cornell, Fall 2021)

Figure 2: k-fold cross-validation

k-fold cross-validation

In k-fold cross-validation we do the following:

- Create a grid of λs
- For each λ:
  - Split data into k mutually-exclusive folds of about equal size, usually choose $k = 5, 10$
  - For $j = 1, \dots, k$
    - fit the model using all folds but fold j
    - Predict out-of-sample on fold j
  - Compute average mean squared prediction error across the k folds: $\bar{Q}(\lambda) = \frac{1}{k}\sum_{j=1}^{k}\sum_{i\in\text{fold j}}\left(y_i - (\alpha_0 + x_i'\beta)\right)^2 + \lambda||\beta||_1$
- Choose $\hat{\lambda}_{min} = argmin_{\lambda}\bar{Q}(\lambda)$ or to avoid modest overfitting choose the largest λ such that $\bar{Q}(\lambda) \leq \hat{\lambda}_{min} + \sigma_{\hat{\lambda}_{min}}$ (1 standard deviation rule)