Modern High Dimensional Linear Regression

Alexander Quispe
The World Bank, PUCP
alexander.quispe@pucp.edu.pe

April 3, 2024

Analysis of Variance (ANOVA)

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

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

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

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 :

-

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

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

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

Understanding β_1 via "Partialling-Out"

1. Lets imagine we have the next equation

$$Y = \underbrace{[\beta_1 D + \beta_2' W]}_{Predicted \quad value} + \underbrace{\varepsilon}_{error} \tag{8}$$

   How does the predicted value of Y change if D increases by a unit while W remains unchanged?

2. **Partialling-out operation:** procedure that takes a random variable V and creates a "residual" $\widetilde{V}$ by subtracting the part of V that is linearly predicted by:

$$\widetilde{V} = V - \gamma_{VW}' W, \;\; \gamma_{VW} = \arg\min_{\gamma} E(V - \gamma' W)^2 \tag{9}$$

3. We can show that

$$Y = V + U \implies \widetilde{Y} = \widetilde{V} + \widetilde{U} \tag{10}$$

Understanding β_1 via "Partialling-Out"

1. Partialling-out to both sides of our regression equation $Y = \underbrace{[\beta_1 D + \beta_2' W]}_{Predicted \quad value} + \underbrace{\varepsilon}_{error}$, we get:

$$\widetilde{Y} = \beta_1 \widetilde{D} + \beta_2' \widetilde{W} + \widetilde{\varepsilon} \tag{11}$$

which simplifies to :

$$\widetilde{Y} = \beta_1 \widetilde{D} + \varepsilon, E[\varepsilon \widetilde{D}] = 0 \tag{12}$$

   - $\beta_2' W \quad \widetilde{W} = 0$
   - $\widetilde{\varepsilon} = \varepsilon$
   - $E[\varepsilon \widetilde{D}] = 0$; since $\widetilde{D}$ is a linear function of $X = (D, W)$
2. What we found in (16) are the Normal Equations for the population regression of $\widetilde{Y}$ on $\widetilde{D}$.

Understanding β_1 via "Partialling-Out"

Theorem (Frisch-Waugh-Lovell, FWL)

The population linear regression coefficient β_1 can be recovered from the population linear regression of $\widetilde{Y}$ on $\widetilde{D}$.

$$\beta_1 = \arg\min_{b_1} E(\widetilde{Y} - \beta_1 \widetilde{D})^2 = (E\widetilde{D}^2)^{-1} E\widetilde{D}\widetilde{Y}, \quad E\widetilde{D}^2 > 0 \tag{13}$$

Theorem (Frisch-Waugh-Lovell, FWL)

Follow the next algorithm:

- *Regress Y on W, obtain residuals ε_1*
- *Regress D on W, obtain residuals ε_2*
- *Regress ε_1 on ε_2, obtain OLS estimates β_1*

Understanding β_1 via "Partialling-Out"

Theorem (Frisch-Waugh-Lovell, FWL)

*In otherwords, β_1 can be interpreted as a (univariate) linear regression coefficient in the linear regression of **residualized** Y **on residualized** D, where the residuals are defined by **partialling-out the linear effect of** W **from** Y **and** D.*

Understanding $\hat{\beta}_1$ via "Partialling-Out"

Theorem (Frisch-Waugh-Lovell, FWL-IN SAMPLE)

When we work with the sample, we mimic the partiallingout in the population.

$$\hat{\beta}_1 = \arg\min_{b_1} \mathbb{E}(\check{Y}_i - \beta_1 \check{D}_i)^2 = (\mathbb{E}\check{D}_i^2)^{-1}\mathbb{E}\check{D}_i\check{Y}_i \tag{14}$$

where $\check{V}_i$ denote the residual left after predicting V_i with controls W_i in the sample:

$$\check{V}_i = V_i - \hat{\gamma}_{VW}' W_i, \;\; \hat{\gamma}_{VW} = \arg\min_{\gamma} \mathbb{E}_n(V_i - \gamma' W_i)^2 \tag{15}$$

Understanding β_1 via ”Partialling-Out”

1. Frisch-Waugh-Lovell-Python
2. Frisch-Waugh-Lovell-Julia

Regression in a High-Dimensional Setting / LASSO

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

$$\min_{b \in \mathbb{R}^p} \sum_i (Y_i - b'X_i)^2 + \lambda \cdot \sum_{j=1}^{p} |b_j| \tag{16}$$

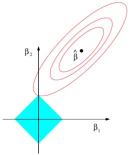
1. The first term is n times the sample mean square error
2. The second term is a penalty term, which penalizes the size of coefficients b_j by their absolute values times the penalty level λ
   A crucial point is the choice of the penalization parameter λ.
3. A theoretically valid choice is (Belloni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

$$\lambda = 2.c\widehat{\sigma}\sqrt{2nlog(2p/\gamma)},\ \widehat{\sigma} \approx \sigma = \sqrt{E \in^2} \tag{17}$$

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:

$$\widehat{\beta}_j = 0 \;\; if \;\; \left| \partial_{b_j} \sum_i (Y_i - \widehat{\beta}' X_i)^2 \right| < \lambda \tag{18}$$

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**

$$\widetilde{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \sum_i (Y_i - X_i'\beta)^2 : \beta_j = 0 \;\; if \;\; \widehat{\beta}_j = 0 \;\; for \;\; each \;\; j \tag{19}$$

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

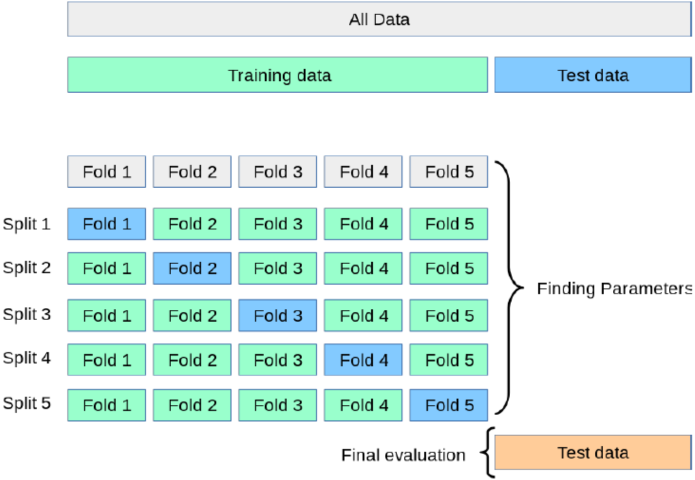
k-fold cross-validation

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 $j = 1, \ldots, k$
    - 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)

k-fold cross-validation

Figure 3: k-fold cross-validation



Practical Value of CEF

1. Randomly split your sample with N observations into a training set with N_{tr} and a test set with N_{te} observations.
2. Estimate different CEF functions in the training sample $\hat{m}(X)$.
3. Calculate and compare their out-of-sample Mean Squared Error (MSE) in the test sample:
$$MSE_{te} = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} (Y_i - \hat{m}(X_i))^2$$
4. Pick the function with the lowest MSE.