Advanced Microeconometrics

Exercise Set 4: High Dimensional Models I

Andreas P. A. Pedersen, Week 40 2025

UNIVERSITY OF COPENHAGEN

Today's Plan

- High Dimensional Setting
- OLS
- Lasso
- Standardization
- Tuning: choice of penalty
- Your time to shine!

High Dimensional Models

- Linear Model:
$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \tag{1}$$
Where $\mathbf{Y}$ is $N \times 1$, $\mathbf{X}$ is $N \times p$ and β is $p \times 1$
- Usually we consider scenarios where N » p and where asymptotics are derived for $N \rightarrow \infty$ such that $\frac{p}{N} \rightarrow 0$
- In high dimensional settings $\frac{p}{N}$ is non-neglible

OLS poor prediction

- Suppose our main aim is to predict **Y** given a set of covariates **X**
- If $\frac{p}{N}$ is non-neglible OLS will perform poorly as the (out of sample) prediction error is

$$E(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{X}_i\hat{\beta} - \mathbf{X}_i\beta)^2) = \frac{\sigma^2 p}{N} \tag{2}$$

  where σ^2 is the variance of the IID error term ϵ
- NB we have moved to a "machine learning world" where prediction rather than causality is the primary aim
- This assumes that OLS is defined. For p $>$ N the rank condition fails since rank($\mathbf{X'X}$) $=$ N $<$ p

Lasso

- Rescue comes from believing sparsity applies i.e. only subset J < p of β are non-zero
- The Lasso estimator perform variable selection and regularization

$$\hat{\beta}(\lambda) = \underset{b \in \mathbb{R}^p}{argmin} \frac{1}{N} \sum_{i=1}^{N} (Y_i - \mathbf{X}_i b)^2 + \lambda ||b||_1 \tag{3}$$

$$\text{where } ||b||_1 = \sum_{j=1}^{p} |b_j| \tag{4}$$

- You can implement the Lasso for given penalty levels λ using the sklearn.linear_model.Lasso package (where α is the penalty level divided by 2 - hence you need to divide your λ-values by 2

Standardize Data

- Lasso is sensitive to the scaling of variables. Variables with bigger standard deviations (represented by larger values) will be penalized more
- To avoid this unintended effect on vartiable selection performed by Lasso standardize the regressors
- We bring them all on the same scale by

$$\tilde{\mathbf{X}} = \frac{\mathbf{X} - \overline{\mathbf{X}}}{\sigma_X} \tag{5}$$

  where $\overline{X}$ and σ_X are the mean and standard deviation of $\mathbf{X}$, respectively
- What does this imply for the interpretation of β?

Tuning: Penalty Selection

- We need to pick a penalty level λ which will be a key variable selection performed by Lasso
- Methods for selecting λ in question:
  - Cross Validation (CV)
  - Bickel-Ritov-Tsybakov Rule (BRT)
  - Belloni-Chen-Chernozhukov-Hansen Rule (BCCH)

Cross Validation

- Divides sample into K subsamples of equal size
- For each subsample K = 1,...K
  - CV uses subsample k for validation and the remaining subsamples for training
  - Computes fit $F_k(\lambda) = \frac{1}{N-M}\sum_{i=M+1}^{N}(Y_i - \mathbf{X}_i\hat{\beta}(\lambda))^2$ where M is the number of observation in each subsample k
- The CV penalty level:

$$\hat{\lambda}^{CV} = \underset{\lambda}{argmin} \sum_{k=1}^{K} F_k(\lambda) \tag{6}$$

- Implementation: sklearn.linear_models.Lasso.LassoCV(cv=K) where K is the number of folds

Bickel-Ritov-Tsybakov Rule (BRT)

- BRT makes two assumptions:
  1. ϵ is independent of **X** and homoskedastic
  2. variance standard deviation σ of y is callable
- To compute $\hat{\lambda}^{BRT}$:
  1. choose $\alpha \in (0,1)$, say α = 0.05
  2. choose c>1, typically c = 1.1
  3. use

$$\hat{\lambda}^{BRT} = \frac{2c\sigma}{\sqrt{N}}\Phi^{-1}(1-\frac{\alpha}{2p})\sqrt{\max_{1\leq j\leq p}\frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_i^2} \tag{7}$$

  where ϕ is the standard normal CDF

- What happens to this formula when we standardize our **X**?

Belloni-Chen-Chernozhukov-Hansen Rule (BCCH)

- BCCH allows heteroskedasticity and requires no preliminary knowledge of the variance of the error terms. (Hint for implementation: Use matrix multiplication)
- To compute $\hat{\lambda}^{BCCH}$:
  1. Choose $\alpha \in (0,1)$ and c as for BRT
  2. obtain pilot Lasso $\hat{\beta}(\hat{\lambda}^{pilot}) = \hat{\beta}^{pilot}$ where

$$\hat{\lambda}^{pilot} = \frac{2c}{\sqrt{N}}\Phi^{-1}(1 - \frac{\alpha}{2p})\sqrt{\max_{1\leq j\leq p}\frac{1}{N}\sum_{i=1}^{N}\mathbf{X}'^2_i(Y_i - \overline{Y})^2} \tag{8}$$

  3. obtain residuals $\hat{\epsilon}_i = Y_i - \mathbf{X}_i\hat{\beta}^{pilot}$ from pilot-Lasso and compute the penalty:

$$\hat{\lambda}^{BCCH} = \frac{2c}{\sqrt{N}}\Phi^{-1}(1 - \frac{\alpha}{2p})\sqrt{\max_{1\leq j\leq p}\frac{1}{N}\sum_{i=1}^{N}\mathbf{X}'^2_i\hat{\epsilon}^2_i} \tag{9}$$

Your time to shine!

- Hint 1: Take a look at the documentation for the following functions: sklearn.linear_model.Lasso, sklearn.linear_model.LassoCV and sklearn.preprocessing.PolynomialFeatures
- Hint 2: These functions are also represented in the lecture slides with additional information
- Hint 3: Methods such as .predict_, .alpha_ and .coef_ are applicable when computing residuals