

ECON-C4200 - Econometrics II

Lecture 7: Machine learning and econometrics

Otto Toivanen

Learning outcomes

- At the end of lecture 7, you
 - 1 understand what **Big data** is
 - 2 what **Machine learning** is
 - 3 how to approach **prediction** as opposed to estimation of **(causal) parameters**
 - 4 what a **Ridge** regression is
 - 5 what a **Lasso** regression is
- We utilize material from [Stock, J. H. & Watson, M. W. \(4th Edition\)](#). *Introduction to econometrics*.

What is Big data?

- When people talk about "Big data", they can mean many things:
 - ① Data with very many observations (typically, millions or more)
 - ② Data that contain lots of variables (typically hundreds or thousands).
 - ③ Data that contain what used to be nonstandard elements such text, voice, images.

What is Big data used for?

- There are numerous ways to use Big data, but by far, the most common usage is prediction in one form or the other:
 - ① What ads would you likely want to see (= what products might you buy)?
 - ② How likely are you to repay your mobile phone loan? (Elisa)
 - ③ Is this you? - recognition problems.
- Machine learning tools are also used for causal analysis both by computer scientists (the so-called **D**irected **A**cyclic **G**raph approach; see e.g. [Paul Hunermund's MOOC](#)) and econometricians (see e.g. [Victor Chernozukov's work](#)) but we concentrate on prediction.

What is Machine learning?

- Machine learning: using a computer and big data to "learn" (to predict as well as possible).
- Though the language used in machine learning is different from that used in econometrics, it (largely) builds on tools that are familiar to econometricians.
- A principle used (but not invented in) in machine learning is to divide data into
 - ① a "training set" which is used to estimate the model and
 - ② a "reserve set" (validation data / Out of sample data) which is used to compare the performance of different models.
- The objective is to predict as well as possible in the reserve set of data.
- **Caveat:** Machine learning utilizes a large variety of tools. We only cover a couple here which are directly based on regression.

Prediction vs. parameter estimation

- Letting go of the objective of unbiased and consistent parameters we can let go of the assumption $\mathbb{E}[\epsilon|\mathbf{X}] = 0$.
- But we need a new assumption: Since we are using one (part of) data to estimate the model and then predict the outcomes in another (part of) data, those need to be similar (enough).
- The latter is called **Out-of Sample** (OOS) data (but also testing data, validation data).
- \rightarrow we assume that (X^{OOS}, Y^{OOS}) is drawn from the same distribution as the estimation data (X, Y) .

Prediction vs. parameter estimation

- So far we have concentrated on what is needed to get unbiased and consistent estimates of β .
- In prediction, the objective is to get as accurate a prediction of the outcome as possible (in the reserve data); hence we do not care about possible biases any longer.
- However, now we need a new benchmark for what is good.
- Enter the "Oracle" who predicts as well as is possible.

Mean Squared Prediction Error

- To be more practical, let us define the **Mean Squared Prediction Error (MSPE)** as:

$$MSPE = \mathbb{E}[Y^{OOS} - \hat{Y}(X^{OOS})]^2$$

- Y^{OOS} = outcome in the reserve / OOS-data.
- X = the variables used for prediction.
- Notice: we estimate the model using the training data, then use the predictors (X^{OOS}) in the OOS data to predict the outcome Y^{OOS} .
- Notice how MSPE is close but different from MSE.

Oracle prediction

- The **Oracle prediction** is the prediction that minimizes MSPE.
- What is this in practice?

$$Y^{Oracle} = \mathbb{E}[Y^{OOS} | X^{OOS}]$$

- Why is this? Imagine this was not the case. Then we could predict the forecast error using X^{OOS} in which case the Oracle prediction could not have been the best possible one.

Standardized regression

- A **standardized** version of variable X is one that has
 - ① mean zero and
 - ② standard deviation of one.
- It is standard in machine learning to use standardized explanatory variables.
- It is also standard to use the **demeaned** version of Y , i.e., $Y - \bar{Y}$.
- → no constant needed.
- Note: if we were interested in the coefficients, they would measure the impact on Y of a one standard deviation change in X .

Standardized regression

- A **standardized** regression has the same form as a regular regression:

$$Y = \mathbf{X}\beta + \epsilon$$

- This allows the use of the same "tricks" as the regular regression: polynomials, logs, interactions, ...
- **Important:** many machine learning methods (including those we cover) depend on what variables the researcher initially "proposes": e.g., they do not on their own start to introduce higher orders of a polynomial.
- By using those tricks you can increase the number of explanatory variables k to the point where $k > n$, i.e., you have more explanatory variables than observations.
- OLS does not work if $k > n$ (the rank condition is not satisfied, i.e., you have more unknowns (parameters) than you have equations).

Prediction error

- Let's write the **true** regression as

$$Y^{OOS} = \beta_1 X_1^{OOS} + \dots + \beta_k X_k^{OOS} + \epsilon$$

- After having estimated the model with the training data we obtain $\hat{\beta}$ and can calculate

$$\hat{Y}^{OOS} = \hat{\beta}_1 X_1^{OOS} + \dots + \hat{\beta}_k X_k^{OOS}$$

- The prediction error has two sources:

- 1 The error term ϵ^{OOS}
- 2 The estimation error in the parameters (coefficients): $\beta_k - \hat{\beta}_k$.

Prediction error

- Let's denote the variance of the error term in the OOS data with $\mathbb{E}[\epsilon^{OOS}] = \sigma_\epsilon^2$.
- This source of prediction error will be there even if we estimate the parameters exactly, i.e., $\beta_k - \hat{\beta}_k = 0$ for all k .
- $\rightarrow \sigma_\epsilon^2$ is the MSPE of the Oracle forecast.
- What is the prediction error of our estimated model?

$$Y^{OOS} - \hat{Y}^{OOS} = (\beta_1 - \hat{\beta}_1)X_1^{OOS} + \dots + (\beta_k - \hat{\beta}_k)X_k^{OOS} + \epsilon^{OOS}$$

MSPE

- The MSPE of the estimated model is then

$$MSPE = \sigma_{\epsilon}^2 + [(\beta_1 - \hat{\beta}_1)X_1^{OOS} + \dots + (\beta_k - \hat{\beta}_k)X_k^{OOS}]^2$$

- For OLS, MSPE is approximately given by

$$MSPE_{OLS} \simeq \left[1 + \frac{k}{n}\right] \sigma_{\epsilon}^2$$

- OLS has a problem: fixing the sample size n , MSPE is increasing in the number of predictors k .
- → need for an estimator for which MSPE increases at a slower rate.

The principle of **Shrinkage**

- It has been known for a long time (since 1950s) that by allowing bias, one can reduce $MSPE$ compared to $MSPE_{OLS}$.
- With uncorrelated X s, these estimators produce coefficients of the form

$$\hat{\beta}^{JS} = c\hat{\beta}$$

where $JS = \text{James-Stein}$ ¹ and $0 < c < 1$.

- The Principle of Shrinkage says that we can reduce MSPE by biasing the coefficients towards zero, i.e., **shrinking** them.

¹ see James, W. & Stein, C. (1961). Estimation with quadratic loss. *Proc. Fourth Berkeley Symp. Math. Statist. Prob.*, 1, 361–379.

The bias-variance tradeoff

- One can show that in estimation, there is a trade-off between variance (of the prediction) and bias.
- This has been known for a long time, but has gained (even) more prominence with machine learning, due to its emphasis on prediction (see e.g. [Giorgos Papachristoudis](#)).
- One can show that we can rewrite the prediction error as

$$MSPE = \mathbb{E}[Y - \hat{f}(X^{OOS})]^2 = \text{bias}[\hat{f}(X^{OOS})]^2 + \text{variance}[\hat{f}(X^{OOS})] + \sigma_\epsilon^2$$

where $f(X^{OOS})$ is our model, e.g.,

$$f(X^{OOS}) = \beta_1 X_1^{OOS} + \dots + \beta_k X_k^{OOS} + \epsilon$$

The **bias-variance tradeoff**

- What happens when you increase shrinkage, i.e., decrease c ?
- You increase bias by definition $\hat{\beta}^{JS} - c\hat{\beta}$.
- At the same time, variance (of the prediction) decreases.
- With a large number of predictors k , the decrease in variance can outweigh the increase in bias.
- This would lead to a lower MSPE.
- The estimators we cover all rely on the shrinkage principle.

Estimating a machine learning model

- 1 Split your data into **estimation** (training) and **testing** data.
- 2 Choose your model.
- 3 Estimate your model with the estimation (training) data.
- 4 Calculate the predicted values of Y , \hat{Y} , for the testing data
- 5 Calculate MSPE using the testing data by summing up the squares of the prediction errors ($Y - \hat{Y}$) and dividing by n_{test} , the number of observations in the testing data.
- 6 Go back to step #2 and repeat until you cannot decrease MSPE no more.

Estimation of MSPE with **cross-validation**

- Best practice is to use k -fold (SW call this m -fold) **cross validation**.
- Example $m = 10$: Divide data into 10 equally sized subsamples.
- Estimate the data leaving one subsample out.
- Predict for the subsample you left out.
- Leave next subsample out, repeat.
- Repeat until you've predicted for all subsamples.
- Sum up the subsample MSPEs to get the MSPE of your estimator.

Machine learning estimators in this course

- We will cover two shrinkage - based regression approaches commonly used in machine learning:
 - ① **Ridge** regression
 - ② **Least Absolute Selection and Shrinkage Operator (Lasso)**.
- Both **penalize** some coefficients, but do this differently.
- We use the empirical examples in SW (ch.14, newest edition).

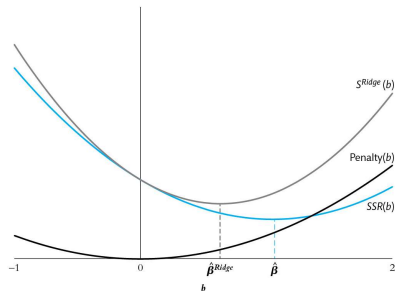
The Ridge regression

- The principle of ridge regression is to penalize coefficients with large squared values by minimizing the following objective function:

$$S^{Ridge}(b; \lambda_{Ridge}) = \sum_i [Y - (\beta_1 X_1 + \dots + \beta_k X_k)]^2 + \lambda_{Ridge} \sum_j b_j^2 \quad (1)$$

- The second sum $\sum_j b_j^2$ is over the coefficients b_j , $j = 1, \dots, k$.
- The term $\lambda_{Ridge} \sum_j b_j^2$ is the **penalty** term.
- One can show that if the regressors are uncorrelated, the ridge regression coefficients take the James-Stein form.

Ridge regression with $k = 1$



Copyright © 2019, 2016, 2013 Pearson Education Inc. All Rights reserved.

- The figure is for $k = 1$. $SSR(b)$ = unpenalized residual sum of squares; $S^{Ridge}(b)$ = the ridge MSPE (objective fcn.)

Lasso

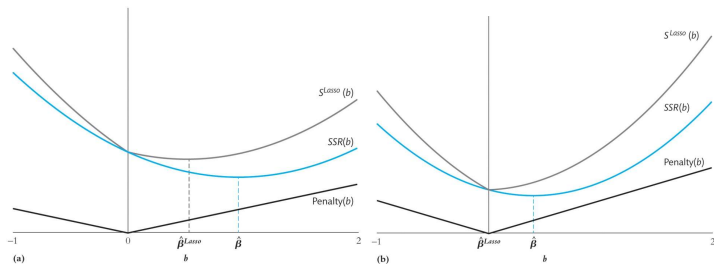
- While Ridge penalizes large values of squared coefficients, Lasso penalizes large coefficients with large **absolute** values:

$$S^{Lasso}(b; \lambda_{Lasso}) = \sum_i [Y - (\beta_1 X_1 + \dots + \beta_k X_k)]^2 + \lambda_{Lasso} \sum_j |b_j| \quad (2)$$

- The second sum $\sum_j |b_j|$ is over the coefficients b_j , $j = 1, \dots, k$.
- The term $\lambda_{Lasso} \sum_j b_j^2$ is the **penalty** term.
- Thus Lasso and ridge very similar in appearance.

Lasso

- Lasso treats large and small coefficients differently:
- It shrinks large coefficients somewhat - less than ridge.
- It shrinks small coefficients to **zero**.
- Lasso therefore in essence "tries to get rid" of variables that affect the outcome only a little.
- A model is **sparse** if most of the true β s are zero and therefore $\mathbb{E}[Y|X]$ really depends on only a few of the X s.
- Lasso seeks to produce a sparse model and therefore works well when the true model is sparse.
- This is a wonderful property when $k \gg n$, i.e., you have many more regressors than you have observations.

Figure: LHS: large coefficients, RHS: small coefficients

Copyright © 2019, 2016, 2013 Pearson Education Inc. All Rights reserved.

Principal components

- Stock and Watson also cover **principal components regression**.
- Briefly, the idea is to "shrink" the number of regressors using so-called **principal components** analysis.
- Once $k < n$, OLS can be used.

SW empirical example: Predicting school level test scores

- Data: school level data on California elementary district data set with additional variables describing the schools, the students and the districts.
- There are 3 932 observations: Half of the (1 966) are used for out-of-sample prediction.
- The data has 817 predictors.

SW school level test score data

Main variables (38)

Fraction of students eligible for free or reduced-price lunch
Fraction of students eligible for free lunch
Fraction of English learners
Teachers' average years of experience
Instructional expenditures per student
Median income of the local population
Student-teacher ratio
Number of enrolled students
Fraction of English-language proficient students
Ethnic diversity index

Ethnicity variables (8): fraction of students who are American Indian, Asian, Black, Filipino, Hispanic, Hawaiian, two or more, none reported
Number of teachers
Fraction of first-year teachers
Fraction of second-year teachers
Part-time ratio (number of teachers divided by teacher full-time equivalents)
Per-student expenditure by category, district level (7)
Per-student expenditure by type, district level (5)
Per-student revenues by revenue source, district level (4)

+ Squares of main variables (38)

+ Cubes of main variables (38)

+ All interactions of main variables ($38 \times 37/2 = 703$)

Total number of predictors = $k = 38 + 38 + 38 + 703 = 817$

SW empirical example: Predicting school level test scores

- Three sets of predictors are used:
 - ① Small - $k = 4$: Student-teacher ratio, median local income, teacher's avg. years of experience, instructional expenditures / student.
 - ② Large - $k = 817$: See the table.
 - ③ Very large - $k = 2065$: Additional school and demographic variables, squares, cubes, interactions.
- For the Very large data set, $k > n$.

Results

Predictor Set	OLS	Ridge Regression	Lasso	Principal Components
Small ($k = 4$)				
Estimated λ or p	—	—	—	—
In-sample root MSPE	53.6	—	—	—
Out-of-sample root MSPE	52.9	—	—	—
Large ($k = 817$)				
Estimated λ or p	—	2233	4527	46
In-sample root MSPE	78.2	39.5	39.7	39.7
Out-of-sample root MSPE	64.4	38.9	39.1	39.5
Very large ($k = 2065$)				
Estimated λ or p	—	3362	4221	69
In-sample root MSPE	—	39.2	39.2	39.6
Out-of-sample root MSPE	—	39.0	39.1	39.6

1. OLS gets worse with more predictors – and you can't even run OLS when $k > n$

Results

Predictor Set	OLS	Ridge Regression	Lasso	Principal Components
Small ($k = 4$)				
Estimated λ or p	—			
In-sample root MSPE	53.6			
Out-of-sample root MSPE	52.9			
Large ($k = 817$)				
Estimated λ or p	—	2233	4527	46
In-sample root MSPE	78.2	39.5	39.7	39.7
Out-of-sample root MSPE	64.4	38.9	39.1	39.5
Very large ($k = 2065$)				
Estimated λ or p	—	3362	4221	69
In-sample root MSPE	—	39.2	39.2	39.6
Out-of-sample root MSPE	—	39.0	39.1	39.6

2. The cross-validated MSPE, computed with the estimation sample, is a good estimate of the out-of-sample MSPE

Results

Predictor Set	OLS	Ridge Regression	Lasso	Principal Components
Small ($k = 4$)				
Estimated λ or p	—	3. Lasso, Ridge, and PC all provide big improvements over OLS		
In-sample root MSPE	53.6			
Out-of-sample root MSPE	52.9			
Large ($k = 817$)				
Estimated λ or p	—	2233	4527	46
In-sample root MSPE	78.2	39.5	39.7	39.7
Out-of-sample root MSPE	64.4	38.9	39.1	39.5
Very large ($k = 2065$)				
Estimated λ or p	—	3362	4221	69
In-sample root MSPE	—	39.2	39.2	39.6
Out-of-sample root MSPE	—	39.0	39.1	39.6

Results

Predictor Set	OLS	Ridge Regression	Lasso	Principal Components
Small ($k = 4$)				
Estimated λ or p	—			
In-sample root MSPE	53.6			
Out-of-sample root MSPE	52.9			
Large ($k = 817$)				
Estimated λ or p	—	2233	4527	46
In-sample root MSPE	78.2	39.5	39.7	39.7
Out-of-sample root MSPE	64.4	38.9	39.1	39.5
Very large ($k = 2065$)				
Estimated λ or p	—	3362	4221	69
In-sample root MSPE	—	39.2	39.2	39.6
Out-of-sample root MSPE	—	39.0	39.1	39.6

4. For these data, Ridge, Lasso, and PC have very similar out-of-sample MSPEs – however this will not be true in general.

- For these data, Ridge has a very slight edge

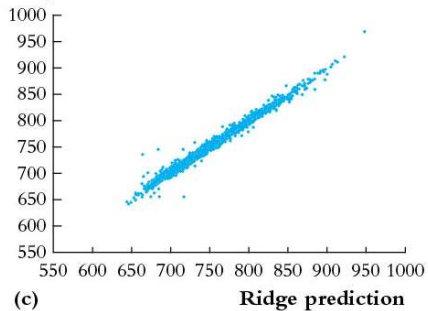
Results

Predictor Set	OLS	Ridge Regression	Lasso	Principal Components
Small ($k = 4$)				
Estimated λ or p	—			
In-sample root MSPE	53.6			
Out-of-sample root MSPE	52.9			
Large ($k = 817$)				
Estimated λ or p	—	2233	4527	46
In-sample root MSPE	78.2	39.5	39.7	39.7
Out-of-sample root MSPE	64.4	38.9	39.1	39.5
Very large ($k = 2065$)				
Estimated λ or p	—	3362	4221	69
In-sample root MSPE	—	39.2	39.2	39.6
Out-of-sample root MSPE	—	39.0	39.1	39.6

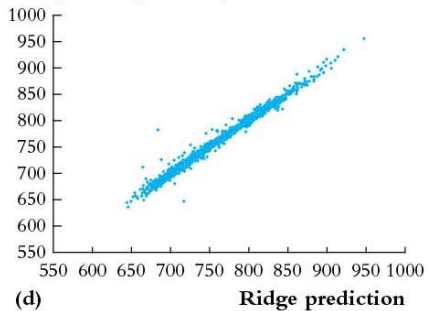
5. For these data, there isn't much gain to using the very large data set, however this will not be true in general.

Results

Lasso prediction



Principal components prediction



Copyright © 2019, 2016, 2013 Pearson Education Inc. All Rights reserved.

Summary

- Many machine learning tools are based on regressions.
- They are designed for prediction, not unbiased estimation of coefficients of interest.
- Machine learning tools are especially useful when there is a large number of predictors / regressors / explanatory variables relative to the size of the data (and one only cares about prediction).
- Ridge and Lasso both utilize the shrinkage principle which builds on the bias-variance tradeoff.
- They easily outperform OLS in prediction in most cases and produce smaller MSPE.