Statistical Learning, Fall 2019

Homework Assignment 1

The purpose of this homework is to apply linear regression on a real data set. We will consider three ways of improving prediction accuracy: feature selection, ridge regression and the lasso. We will split the data set into two parts (training and test set), then fit the models on the training set (including choice of the best parameter using cross-validation) and test their performance on the separate test set.

General Remarks

- Please adopt an academic writing style, i.e. "model A outperforms model B on criteria Z by a factor X, which can be explained by argument Y (reference)" instead of ~~"we think the model's performance is impressive and it obviously does better than the other thing as our teacher already explained"~~.
- That said, we welcome you to explain aspects in your results that surprised you or from which you have learned. Include comments when necessary, both in code as well as in writing. You are also encouraged to include plots presenting some of the results.
- It is allowed to compute results with either R or Python, but we advise R given the packages we suggest throughout the assignment.
- Although code is a necessity for this report and its results a significant contributor to the grade, it is not sufficient. The majority of points is earned by interpreting and explaining your findings in a separate report.
- We encourage students learning from one another, but make sure to write your code and your report individually.

Exercises

Introductory analyses

1. Download the `housing` data set, concerning housing values in the suburbs of Boston. The detailed explanation what the input and output variables mean, can be found here.
2. Read the data set (for R: `read.table`, for python: `pandas.read_table`). Divide the set into two parts. Use the first 350 examples as a training set and the rest (156 examples) as a test set.
3. Fit the least squares model to the training set (for R: `lm` R hint: to specify that `MEDV` is the output variable, while all the others are input variables, you can use formula '`MEDV` $\sim$.', for python: follow this brief example).
4. Report estimated coefficients, their standard errors, and statistical significance (in R you can get these by using `summary`, for Python see the example mentioned above). If you use statistical significance to explain the differences between the different models in this assignment make sure you check the assumptions (for a review of these assumptions, see source).

Cross-validation and regularization

Hint: you might want to study §7.10.2 in The Elements of Statistical Learning.

5. Estimate the prediction accuracy of regression using 5-fold cross-validation. You have to write the code for cross-validation yourself. Compare the cross-validation error with the residual sum of squares ("training error"). Please interpret and explain your findings.

6. Perform full subset-search. In R you can use the function `regsubsets` from package `leaps`. The function returns for $k = 1, \ldots, p$ the best subset of size k in terms of mean squared error on the training set. In python there is no easy way to do full subset search, but this page could be useful: subset selection in python. By cross-validation, choose the best k (in R, to access the subsets, you can use `summary(regsubsets.output)$which`, where `regsubsets.output` is the output of the `regsubsets` function).

7. Apply ridge regression to the training set. In R you can use the function `lm.ridge` from package `MASS`. In python you can use the `sklearn` package. Choose several values of shrinkage and get the best one using cross-validation (in R you can use either `select` procedure for `lm.ridge` output, which uses "approximated" cross-validation (GCV), or you can use your own cross-validation method). Then predict outcomes for the test set (to be used in question 9). Hint: make sure to use unnormalized coefficients (assuming you have left the test data unnormalized) from the training procedure. In R `coef(ridge.model)` gives unnormalized coefficients, whereas `ridge.model$coef` gives normalized coefficients. Furthermore, note that the coefficients include an intercept, so introduce this in your test data as well.

8. Apply lasso regression to the training set. In R you can use functions `lars` and `cv.lars` with option `type="lasso"` to fit the lasso model and choose best shrinkage with cross-validation. Use `predict.lars` for prediction. In python you can use the `sklearn` package.

Model comparison

9. Compare all four models (least squares regression, best-subset, ridge, and lasso) resulting from optimally tuned parameters (subset size or shrinkage) on the test set. Report prediction accuracies and draw conclusions. Please make sure to at least address the following points:

   (a) Comment on the magnitude of the differences.

   (b) Make an informed comparison against the least squares results, i.e. try to connect the theory to the results you have found.

   (c) Is any of the models useful to predict the outcome of housing prices? Explain why.

   (d) Compare the test-set error against the CV-error. Explain your findings.

Please submit your code (e.g .R or .py) and report (in .pdf format) including your name and student number to blackboard (under information).