UIO

FYS-STK4155
Project 1

October 10th, 2020

Kjell R. Christensen

ABSTRACT

Machine learning as a research field, has for the last decade bounced back, “again”, due to waste amount of data that is being generated. Also adding to the uptick is the enormous processing power, some of the new datacenters, like Google and Microsoft can offer, not only the professional, but to the public in general.

Never, has so much data been generated by social medias, banks, insurance, sales organizations, and research is again spread wildly on many different disciplines.

But still, some data is hard to produce, and need planning and huge amount of resources to extract. A good example are experiments done in CERN. It is extremely costly, limited and restricted access and produces huge amount of data, for post processing. Due to the latter, a lot of research within e.g. nuclear and astrophysics might at time be based on limited amounts of datasets. How much data is needed to enable good predictions?

This project will make an analysis on small datasets by using linear regressions. We will compare three common methods (OLS, Ridge & Lasso) and show that one of them is more favorable than the other two, when it comes to regression analysis, for limited numbers of datasets. We are seeking a “lower limit” of data needed, but still being able to generate cost functions that satisfies our pre-defined error tolerance.

INTRODUCTION

The task in hand is to optimize the combination of cost functions and number of datasets needed. Regression methods (OLS, Ridge & Lasso) will be used and we are also optimizing the test and training tasks, by using Bootsrap & K-Fold as resampling methods.

For small amount of data, the split between test and training data can easily be unsymmetrical and, shuffling data and resampling make certain that all test data are unitized as best possible.

Bootstrap and K-Fold will in principle generated a mean error rate based on the number of iterations they do, towards the linear regression methods.

Both Bootsrap and K-Fold routine are extended to wraps around the three regression methods, to make the testing more optimal and give a better overview.

For OLS we are in a search for best polynomial fit, but for Ridge and Lasso, we seek the optimal lamda combination for the data sets.

METHODS

The analyses is done in a msCode environment and is using standard libraries as specified in the *.py heading. All “imports” are based on standard libraries, commonly used for ML, and routines and code generated for this project, is located only in the python file.

I have use two datasets, one test-set, generated by extracting random numbers within [0,1] and scaling them by subtracting the mean. Second dataset is based on real topographic data. I have been scaling these data in the same manner as for the test-set.

Both datasets are feed into the Franke’s Function f(x,y) and returning a vector z, for further testing.

The program is modularized, based on separate tasks, and I have modularized in such a way that the program itself, can run one, several or all modules, in one go for analyses. A separate test environment is, hence, not needed.

The reader can at any time reproduce any test-results or plots, by activating the selected part in the MainModule().

All plots are located in the ./Plots folder and they are named, based on function, parameters and methods. The naming convention is also indicating the size of the sample data, to ease the setup, and re-run the tests.

Part of the analysis is to benchmark the resampling methods, k-fold and bootstrap.

I have expanded both the k-fold and bootstrap routine for cross-validation to also include the encapsulation of the outer-loop for, polynomials and set of lamdas, respectively. This makes the code more re-usable for the different test-cases.

RESULTS & DISCUSSIONS

For small amount of data sets I found that results were unpredictable and MSE error was bouncing up and down between degrees of polynomials. Also for Ridge and Lasse the lamda had a braking point for lamda=1, where error bounced up. This need to be looked into futher.

For OLS we could see that test data matched the training set quite nicely, when samples passed 200+.

For Ridge and Lasso, the current result is not clear and also need to be explored futher.

CONCLUSIONS & EXPECTATIONS

Our aim for the project was to test and compare three regression methods for limited amounts of datasets. A clear objective was to find the most suitable polynomial for the linear regression method, based on OLS was the first part of the project.

For the OLS method, our prediction model/f(x) is based on a polynomial approximation. Our scope was limited, in the first part and polynomial boundaries was p in range [0,21] giving a maximum polynomial of degree 5. This was not optimal, and several tests gave an indication that best fit most likely would be between 6^{th} and 7^{th} degree.

For the topographical data the OLS fitting look good at polynomial degree 8

and falls a bit short of the optimal degree that is more likely to be between 6 or 7. Since the scope for this part is polynomial degree >=5, we see that the optimal value

Before reaching an absolute conclusion, future work should include more methods for linear regression. One should also include more graphs like heat-maps to visualize the findings and optimal values.

As for the testing done up until now, I find the OLS regression maps best to the data sets and will be favorable to use.

Even though, I have reached an intermediate conclusion, at current time, I don’t see this analysis as finalized and is subject to further studies in the future.

Tasks to look further into:

1) Make a clear granularity for OLS analysis for different sample sizes. It is clear that by increasing the samples, the breaking point for MSE are stretch out on higher polynomials. E.g. for samples size of 100, we get a breaking point around degree 6. For sample 500 the breaking point moves up to degree 7. Further investigation is needed to
2) Ref 1), we also need to break down values and combination for lamdas and samples. It is not a clear cut, where we have the best cross-validation yet.
3) Computational cost vs MSE for higher polynomials. Some tests have not been finalized, due to lack of computational “power”.
4) Bias vs Variance relationship.

REFERENCES

[1] Morten Hjorth-Jensen, Computational Physics, Lecture Notes and sample code, Fall 2020, https://github.com/CompPhysics/MachineLearning

[2] Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Second Edition.
DOI https://**doi**.org/**10.1007**/978-0-387-84858-7; Copyright Information Springer-Verlag New York 2009; Publisher Name Springer, New York, NY; eBook ...

[3Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurelien Geron. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow [2nd ed.] 978-1-492-03264-9. 174 47 66MB. English Pages 1150 Year 2019.

[4]Data Science from Scratch-First Principles with Python, Grus.
Afficher les collections Cacher les collections. Identifiant. ISBN : **978-1-491-90142-7**. ISBN : **978-1-491-90142-7**. ISBN : 1-491-90142-X. ISBN : 1-491-90142-X.

[5] Practical Statistics for Data Scientists - Peter Bruce & Andrew Bruce
978-1-491-95296-2 [M] www.allitebooks.com Dedication We would like to dedicate this book to the memories of our parents Victor G. Bruce and Nancy C. Bruce, ...

[6]Python Machine Learning, Sebastian Raschka & Vahid Mirjalili
23. sep. 2017 — Python Machine Learning - von Sebastian Raschka, Vahid Mirjalili (ISBN **978-1-78712-593-3**) bestellen. Schnelle Lieferung, auch auf ...

[7]A Primer on Scientific Programming with Python, Hans Petter Langtangen
ISBN **978-3-662-49886-6**; Free shipping for individuals worldwide. Please be advised Covid-19 shipping restrictions apply. Please review prior to ordering.