Paper Replication: Empirical Asset Pricing via Machine Learning

MATH5470 final project
Department of Mathematics
HKUST
Jialin ZENG(20807909)
Tianyu JIN(20808264)
Tiankai HU(20808238)
{jlzeng,tjinac,thuah}@connect.ust.hk

Abstract

This report is an reimplement of the paper "Empirical Asset Pricing via Machine Learning"[1]. We reproduce 6 machine learning methods used in this paper, including linear regression(OLS,elastic net), dimension reduction(PLS,PCR), gradient boosting regression tree and neural networks. Details of these models will be introduced from section 2 to section 5. As for the evaluation part, we calculate the out-of-sample R-square scores and analyze the variable importance for each model. Finally we compare the mean R-square scores of different models and obtain the conclusion.

1 Data Preprocessing

We select 60 years of data from 1960-2019 as the entire data, including 18 years(1960-1977) training sample and 12 years of validation sample(1988-1989) and 30 years(1990-2019) of test sample. The data includes three parts: firm characteristic, SIC code and the macroeconomic predictors.

The macroeconomic predictors are given in a different table than the first two parts. Before merging three parts of data together, we first use formulas to calculate out 8 macroeconomic predictors specified in the paper and then we calculate the individual excess return(not RET given in the first table) for regression goal following instruction of the paper.

Due to limited computational resources, we do not calculate Kronecker product between firm predictors and macroeconomic predictors to construct 920 covariates. We believe in neural networks and tree models, such nonlinear interactions between predictors can also be guaranteed. Therefore, we drop the unmentioned predictors and keep the original 103 predictors as final predictors.

The merging data includes a lot of missing values, which will cause bad influence to regression result without filling properly. Therefore, we calculate the cross-sectional median instead of all-time median to fill the missing characteristic so that no future information will be included in the present statistics. For the missing value after this filling, we fill them by zero.

We adopt the recursive performance evaluation scheme mentioned in the paper: When evaluating every new year in the test sample, we let the training sample increase by one year, the validation sample roll it forward by 1 year while keeping the fixed size of 12 years.

2 Linear Regression

The purpose of linear regression is to find a simple linear relationship between the independent variable and the dependent variable. In this section, we will show the results using OLS, OLS and Elastic Net methods. It is easy to know that simple linear regression does not work well for high-dimensional datasets. But we can still use it to generate some prior knowledge about the data.

Here, instead of using standard least squares, we use Huber robust objective function, which is defined as

$$\mathcal{L}_H(\theta)=\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} H\left(r_{i,t+1}-g\left(z_{i,t};\theta\right),\xi\right),$$

where

$$H(x;\xi)=\begin{cases} x^2, & \text{if } |x|\le\xi \\ 2\xi|x|-\xi^2, & \text{if } |x|>\xi \end{cases}$$

2.1 OLS and OLS-3

The OLS is the simplest linear model and the least effective. As in the article, our goal is to use a simple linear model to predict the excess return. We can use a simple linear model to make some prior estimates. So in OLS model we calculate the correlation coefficient of each feature to represent the variable importance.

To do a simple linear regression, the OLS-3 only focuses on using three covariates,'mom12m','bm'and'mvel1' to predict the excess return.The average R^2_{oos} of OLS-3 is -0.0863.

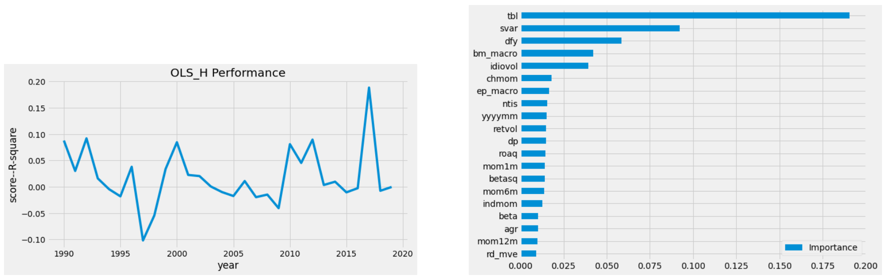
2.2 Penalized Linear Regression-Elastic Network-Huber

Penalized methods differ by appending a penalty to the original loss function:

$$\mathcal{L}_H(\theta;\cdot)=\mathcal{L}_H(\theta)+\phi(\theta;\cdot).$$

Here we use "elastic net" penalty, which takes the form

$$\phi(\theta;\lambda,\rho)=\lambda(1-\rho)\sum_{j=1}^{P}|\theta_j|+\frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2$$

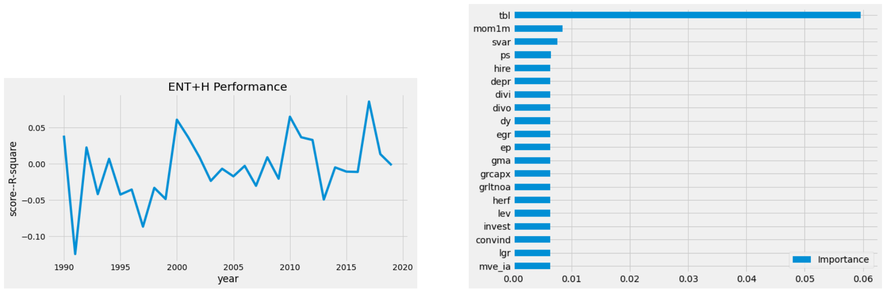


(a) (b)

Figure 1: (a) Annual return forecasting R^2_{oos} of OLS. (b) Top-20 most influential variables for OLS. The average R^2_{oos} of OLS is 0.0176.

The Elastic Net involves two nonnegative hyperparameters, λ and ρ. We use validation datasets to choose the optimal hyperparameters where $\rho = 0.5$ and $\lambda = 0.05$.

Here to calculate the Variable Importance we use a method mentioned in paper, the reduction in panel predictive R2 from setting all values of predictor j to zero, while holding the remaining model estimates fixed. The performance of ENET has shown in Figure2.



(a) (b)

Figure 2: (a) Annual return forecasting R^2_{oos} of ENET. (b) Top-20 most influential variables for ENET. The average R^2_{oos} of ENET is -0.0057.

3 Dimension Reduction: PCR and PLS

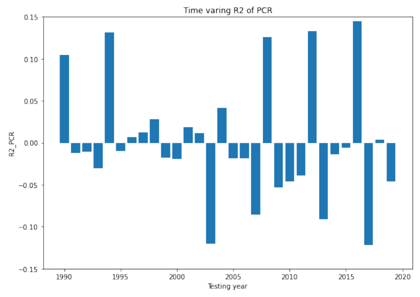
3.1 Model Formulation

Dimension reduction utilizes the idea of predictor averaging. It can decorrelates highly dependent predictors by projecting P-dimensional predictors to a smaller K-dimensional subspace. We replicate two classical dimensional techniques principle components regression(PCR)and partial least squares(PLS). PCR projects predictors to K-dimensional subspace that can best preserve the covariance structure among predictors and performs the linear regression using leading components selected. The drawback is it doesn't consider associating with final predictive goal. In contrast, PLS exploits such correlation between predictors and forecast target. Both these two methods averages noise by combing predictors into low-dimensional components. The forecasting model for both two methods can be written as

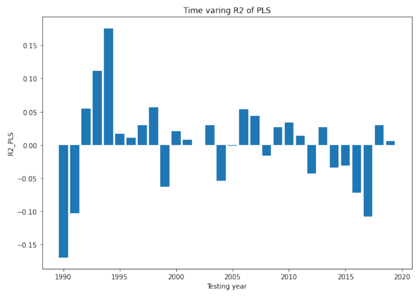
$$R = (Z\Omega_K)\theta_K + E$$

,

where Ω_K is $P \times K$ matrix with columns $\omega_1, \omega_2, \ldots, \omega_K$ and $Z\Omega_K$ is the dimension-reduced version of the original predictor set.

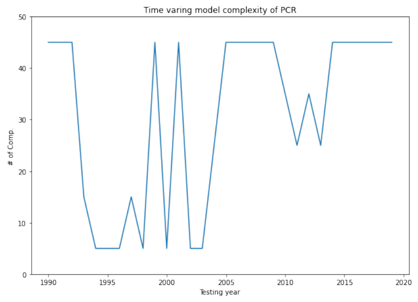


(a)

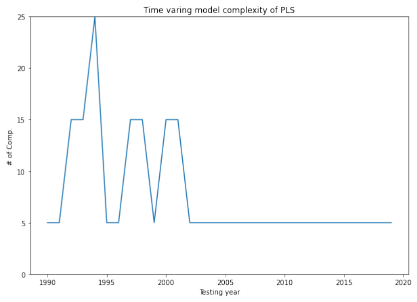


(b)

Figure 3: (a)Time varing R-square scores for PCR with mean R^2_{oos} 0.01904. (b) Time varing R-square scores for PLS with mean R^2_{oos} 0.01531



(a)



(b)

Figure 4: (a)Model Complexity for PCR. 1 (b) Model Complexity for PLS.

PCR looks for each ω_k recursively that maximize the variance of $Z\omega$, so j^{th} linear combination solves

$$\omega_j = argmax_\omega Var(ZW) s.t. \omega^{'}\omega = 1, cov(Z\omega, Z\omega_l) = 0, l = 1, 2, \ldots, j-1.$$

PLS looks for K-dimensional projections of Z that maximize predicative association with the final target R, so the j^{th} linear combination solves

$$\omega_j = argmax_\omega Cov^2(R, ZW), s.t.\omega^{'}\omega = 1, cov(Z\omega, Z\omega_l) = 0, l = 1, 2, \ldots, j-1.$$

3.2 Performance Evaluation

We perform PCR and PLS to the merged data respectively and obtain the yearly R^2 score for the testing sample(1990-2019) as shown in Figure 3. From this result we can see that in general, both methods have better and more stable predictive power in the earlier testing years. This might be due to the fact that stock market is highly time-dependent. Moreover, performance of PCR method is more reliable than PLS in the long run. It may be because PCR focuses more on capturing covariance structure of predictors rather than correlation between forecast return. It seems past correlation between predictors and return has little power to predict the future pattern thus the covariance structure among predictors stands out.

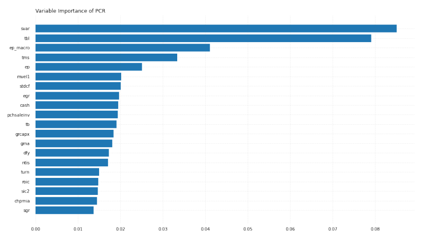
We also perform model complexity analysis to each model, as shown in Figure 4. For each testing year, we select the best hyperparameter, i.e.the number of components K by validating on the validation sample. Due to computational cost, though we only validate component numbers from $[5, 15, 25, 35, 45]$, the result pattern is similar to the one given in paper. That is, PCR typically use 25 to 45 components in its forcast while PLS falis to stablize in the early stage but eventually settles on about 5 components.

For the variable importance analysis, we follow the paper's method to measure each predictors's importance using the R^2 reduction from setting all values of this specified predictor to zero, while holding the remaining model estimates fixed. Again due to limited computational power, we only calculate the R^2 reduction for the testing year with best R^2 performance and corresponding best component numbers. From Figure 5, we can see that the importance magnitude is highly skewed toward many macroeconomic predictors such as tnl, dfy and some firm predictors like mom1m.This is plausible since we independently merge firm predictors and macroeconomic ones rather than conduct product interaction between them. Moreover, the nature of dimension reduction method also leads us to discover just a few leading components.

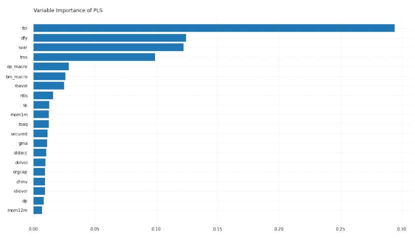
4 Gradient Boosting Regression Tree

Regression trees are the most popular machine learning methods, which are intuitive for decision-making. Unlike linear regression, regression tree is a nonparametric method and quite different from traditional regressions. Our target is to divide the observations into small groups. For each group, they behave similarly.

Boosting is a regularization method, which converts many weak learner into a strong one. Combining with regression trees, we have gradient boosting regression



(a)



(b)

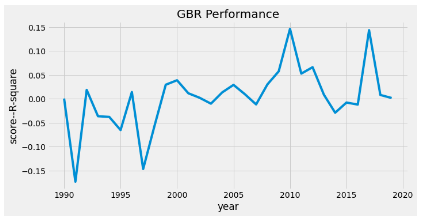
Figure 5: (a)Top-20 most influential variables for PCR. 1 (b) Top-20 most influential variables for PLS.

tree(GBRT). GBRT recursively considers results from trees with simple structure to get a strong learner with higher stability than a single complex tree.

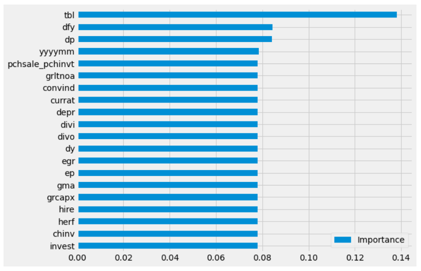
Here, we use the Huber loss as our loss function and the hyperparameters we need to train are the weight reduction factor of each weak learner (step size) and the maximum depth of the individual regression estimators.

We use validation datasets to choose the optimal hyperparameters where $max\_depth = 3$ and $learning\_rate = 0.01$ and we also use Huber loss.

The performance of GBRT are shown in Figure6.



(a)



(b)

Figure 6: (a) Annual return forecasting R^2_{oos} of GBRT. (b) Top-20 most influential variables for GBRT. The average R^2_{oos} of GBRT is 0.0030.

5 Neural Networks

The structure of neural networks used in this paper are based on traditional "feed-forward" networks, which consist of an "input layer" of raw predictors, one or more "hidden layers" that interact and nonlinearly transform the predictors, and an "output layer" that aggregates hidden layers into an ultimate outcome prediction. Here, the input layer and the output layer remain the same from NN1 to NN5, and we only change the width and depth of hidden layers. NN1 has a single hidden layer of 32 neurons. NN2 has two hidden layers with 32 and

16 neurons, respectively. NN3 has three hidden layers with 32, 16 and 8 neurons, respectively. NN4 has four hidden layers with 32, 16 ,8 and 4 neurons, respectively. NN5 has five hidden layers with 32, 16, 8, 4 and 2 neurons, respectively. ReLU is used at all nodes as activation function, and the loss function is $L = 1 - R^2$.

In the training process, we choose SGD as the optimizer. The "learning rate shrinkage" algorithm is used to adjust the learning rate as the gradient approaches zero in the optimization process. Also, we employ a general machine learning regularization tool, "early stopping", for a much lower computational cost. The training process of NN1,NN2,NN4 run about 30 epoches before early stopping, while NN3 and NN5 are trained about 60 epoches.

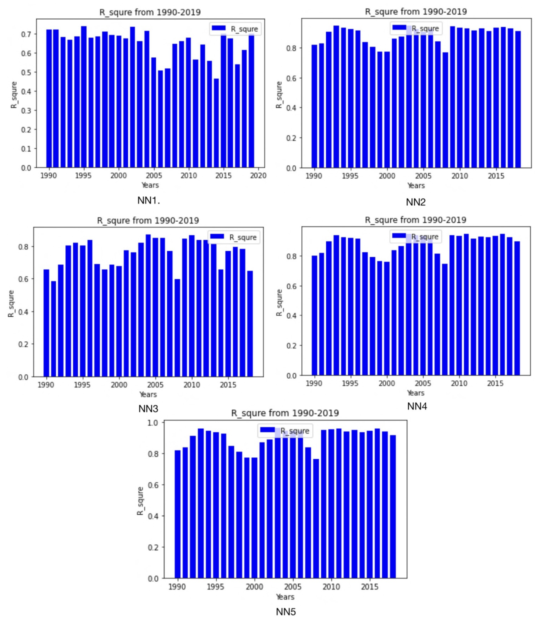


Figure 7: R-square scores for NN1-5(1990-2019)

Figure 7 shows the out-of-sample R-square scores for all test years(1990-2019). The mean R-square score for NN1 is 0.06496, while those for NN2 to NN5 is between 0.075-0.091. This result shows that the performance of neural networks gets a great improvement as we extend the single hidden layer to the depth of 2, but as the depth become larger than 2, the improvement is not obvious.

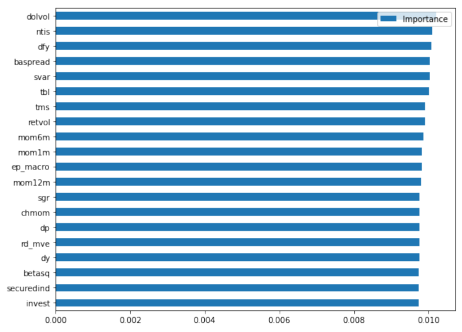


Figure 8: Variable Importance for NN1-5

Figure 8 shows the top 20 average variable importance of NN1-5. The difference between each feature is not obvious. But still, we notice that variables such as dolvol, baspread, retvol, mom6m, mom1m, mom12m, chmom, betasq, securedind, which are said to be important for neural network models in the paper, are also the top-20 most influential variables in our NN1-5.

6 Conclusion

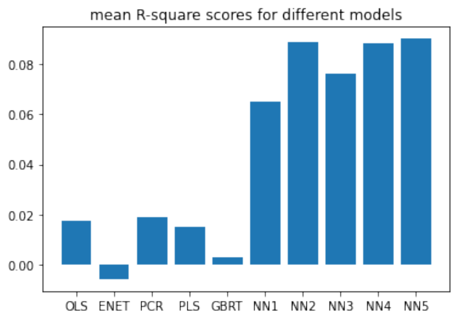


Figure 9: Mean R^2 scores for different models

Figure 9 shows the comparison of R-square scores for different models. As expected, Neural Networks perform best among all models. PCR, OLS-3+H, and PLS are also effective in this task, but their scores are much smaller than NN1-5. ENet has the worst R-square score, which means that it is not effective in this prediction task. We have also obtained several interesting and helpful observations:

- OLS-3 outperforms the OLS model, indicating that we can **averages out noise by reducing the number of predictors**.

- PLS and PCR can help with dimension reduction and **pick out several more important components for further regression**.
- Linear interations between predictors is not enough. Tree model and neural networks can better **boost performance by adding nonlinear interations**.
- **Shallow neural networks seems to outperform deep ones**. It may be due to insufficient data and large proportion noise.
- We discover in general macro predictors like **dfy and tbl** have larger impact on the prediction. For the firm predictors, **momentum variables such as mom6m and mom1m** are more influential.

Therefore, future improvements of our work could be a **combination of linear interations and nonlinear ones**. We can first select important components to reduce dimension, then use more powerful neural networks to make final predictions.

Workload

- Jialing ZENG takes part of Data Preprocessing and Dimension Reducing.
- Tianyu JIN takes part of Neural Network.
- Tiankai HU takes part of Linear Regression and Tree methods
- Also, we thank Chuqi CHEN for the assistance on the part of Linear Regression.

References

[1] Gu, S. & Kelly, B. & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies 33(2020)*, pp. 2223-2273