Risk Management

Homework 2
Option Return Predictability
Deadline April 28th 2021, 23:59

This homework constitutes 15% of the total mark for this course. You should clearly indicate the names of group members.

Submit a report with a detailed description of the analysis. The corresponding Matlab/R/Python code and the data should also be submitted. Make sure that the results can be reproduced with the files you submit.

Send all files to Mykola via email (mykola.babiak@gmail.com). The subject line of your email should be "Med Project BA UCU".

Good Luck!

The main goal of this project is to provide an answer to the question: do machine learning methods explain option returns? The extant literature has documented that it is notoriously difficult to study option prices due to their short lifespans and migrating moneyness. This project will propose, compare, and evaluate a variety of different machine learning methods for predicting equity index option returns and compare the results to the standard linear regression models common for the prior literature.

1. Load the file "SP500_Options_Monthly.h5". Focus on the following characteristics:

- Information including contract specifications (**C/P**, **strike**, **midprice**, etc.)
- Underlying index values, that is, S&P500 index (**spot_close**)
- Historical dividend yields (**divrate**)
- Option sensitivity measures such as the BMS Delta (**delta**), Gamma (**gamma**), Vega (**vega**), and Theta (**theta**).

2. To introduce notation, each option contract i is defined by its strike price K_i and maturity date T_i. Hence, the time-to-maturity (**ttm**) is given as $T_i - t$. As a measure of the position of an option contract i relative to its strike K_i, compute moneyness standardized by implied volatility (**mness**) as

$$m_{i,t} = \frac{\ln(K_i/S_i)}{IV_{i,t}\sqrt{T_i - t}},$$

where $IV_{i,t}$ denotes the BMS implied volatility (**impvol**) of the contract.

3. In your computations, you need to forecast the prices of option contracts (**midprice**)

4. Define embedded leverage (**embed_lev**) as

$$\Omega = |\Delta \cdot S/F|.$$

5. Provide summary statistics (mean, median, standard deviation, number of observations) of **mness, ttm, embed_lev, impvol, gamma, vega, theta** and **midprice** for put and call options

6. Estimate a panel regression model of the following form

$$p_{t,option} = \alpha + \beta X_t + \varepsilon_t,$$

where $p_{t,option}$ is the option price, X_t is a 13×1 vector of the option-level variables

(1) moneyness (**mness**)
(2) time-to-maturity (**ttm**)
(3) embedded leverage (**embed_lev**)
(4) BMS implied volatility (**impvol**)
(5) BMS Gamma (**gamma**)
(6) BMS Vega (**vega**)
(7) BMS Theta (**theta**)
(8) strike (**strike**)
(9) implied volatility (**impvol**)
(10) S&P index (**spot_close**)
(11) dividend yields (**divrate**)
(12) VIX index (**vix**)
(13) short-term interest rates (**short_rate**)

Report the regression output and explain the sign and significance of all estimates.

7. Consider a predictive regression model of the following form

$$p_{t,option} = \alpha + \beta X_t + \varepsilon_t,$$

where $p_{t,option}$ is the option price, X_t is a 13×1 vector of the option-level variables. Evaluate how the following techniques help better predict equity index option prices.

- OLS regressions,

- penalized linear regressions (Ridge, Lasso, Elastic Net),
- principal component analysis (PCA) (3, 5 and 10 components)
- random forests
- boosted regression trees
- extremely randomized regression trees
- neural networks: shallow vs. deep (for example, 1 layer with 16 nodes vs. 2 layers with 16-8 nodes)

Starting from **January 2007 (t_0 = 2007-01)**, perform the following estimation strategy. Following common machine learning practice, you should split the historical data (that is, the data available at the time you train the corresponding model) into two sub-samples: a training set used to train the model and a validation set used to evaluate the estimated model on an independent data set. Use the model accuracy over the validation sample to iteratively search the hyperparameters that optimize the objective function.

Regarding the splitting scheme, apply the following rules:

- keep the **fraction** of data used for training and validation fixed at 85% and 15% of the historical data, respectively
- training and validation samples are consequential, that is, you always take the first 85% of the historical data for training and then the remaining 15% of the historical data for validation while preserving the order of observations. In other words, you do **not** cross-validate by randomly selecting independent subsets of data to preserve the time-series dependence of both the predictors and the target variables.
- Forecasts are produced recursively by using an **expanding** window procedure, that is, we re-estimate a given model at each time t and produce out-of-sample forecasts of excess returns at time $t + 1$. Also, due to the expanding window, we will have more and more historical data and hence larger training and validation samples

- Notice that for some of the methodologies, validation is not required. For instance, neither standard linear regressions nor PCA require a pseudo out-of-sample period to validate the estimates. In these cases, we adopt a traditional separation between in-sample versus out-of-sample period, where the former consists of both the training data and the validation data.
- For neural networks, it might be too computationally costly to fine-tune the networks each month. For simplicity, assume that you reestimate the network each 5 years (or 10 years if cross-validation is very slow and you do not have access to the high end computing cluster). In other words, if you fine-tune the network at time t, then you should use this network without changing the hyperparameters for forecasting excess returns in all periods for the next 5 years. You reestimate the network only in period $t+$ (5 years).

Report the out-of-sample Mean Squared Prediction Error (MSPE) and R^2_{oos} computed as follows:

$$MSPE_s = \frac{1}{T-t_0-1}\sum_{t=t_0}^{T-1}\left(p_{t,price}-\hat{p}_{t,price}\right)^2,$$

$$R^2_{oos} = 1 - \frac{\sum_{t=t_0}^{T-1}\left(p_{t,option}-\hat{p}_{t,option}\right)^2}{\sum_{t=t_0}^{T-1}\left(p_{t,option}-\overline{p}_{t,option}\right)^2},$$

where $\hat{p}_{t,option}$ is the one-step ahead forecast of option prices; $\overline{p}_{t,option}$ is the historical mean price across all options; t_0 is the date of the first prediction. Discuss your results. You may want to answer some of the questions below.

- Compare the predictability implied by tree-based methods and the standard linear regressions.
- How do extreme trees affect the predictability?
- How do shallow and deep neural networks affect the predictability?

Suggest and perform some robustness checks and then discuss how your initial results change.

8. Consider a predictive regression model of the following form

$$p_{t,option} = \alpha + \beta X_t + \varepsilon_t,$$

where $p_{t,option}$ is the option price, X_t^{Int} is a 26×1 vector of the 13 option-level variables X_t and 13 asymmetric terms

$$\mathbb{I}(\text{Contract } i \text{ is a put option}) \times X_t,$$

where

$$\mathbb{I}(\text{Contract } i \text{ is a put option})$$

is an indicator variable that is equal one for put options and equal zero for calls. In other words, we allow for the possibility that characteristics determine option returns differently for calls and puts. Repeat the steps from Part 7.