DSO 530: Statistical Learning Methods
Group Project Description
Spring 2020

This document is to describe the group project of Dr. Xin Tong's DSO 530 class in Spring 2020 at the University of Southern California. In this project, you will be conducting algorithmic trading using supervised learning approaches. You have a choice to propose a different (but equally exciting and challenging) problem to work on. But should you choose that, your group has to discuss with the instructor and get permission by March 26. The project will be completed in groups of (up to) 5 people each. A group of size 6 will receive 20% penalty out of your group grade. No group should have size 7 or above. You can form groups with people across two sessions. If you prefer to do this project alone, it is also allowed. In addition to this project description, you should also read the `technical guidance` document posted on Blackboard. `This project is an independent group project. The instructor and TA will refrain from giving specific advices except during the presentation.`

Background of the problem:

Algorithmic trading executes trade orders using automated trading instructions. This type of trading was developed to make use of the speed and data processing advantages that computers have over human traders. In the twenty-first century, algorithmic trading has been gaining traction with both retail and institutional traders. It is widely used by investment banks, pension funds, mutual funds, and hedge funds that may need to spread out the execution of a larger order or perform trades too fast for human traders to react to. A study showed that over 80% of trading in some key markets was performed by trading algorithms rather than humans.

Other than IT technologies, a center piece of algorithmic trading is prediction models based on past data and statistics and machine learning techniques. Understanding and forecasting financial asset returns has been one of the central topics for researchers across academia and industry, and return prediction is generally formulated as a regression problem. Due to the very nature that all market participants are trying to beat market average and maximize profits, exchange-traded asset prices are quite efficient, and future returns mainly consist of idiosyncratic noise and are hard to predict.

When an investor decides whether to buy a certain asset, she faces fundamental asymmetry between taking action versus not. If she buys and the value of assets goes down, she sustains a real accounting loss; if no action is taken and the value of assets goes up, it incurs only loss of opportunity. Due to limited research capability and funding capacity, there are many more opportunities in the universe of assets than the investor can possibly capture. On the other hand, prospect theory in behavioral economics suggests that for investors as humans, the suffering from a loss greatly outweights the pleasure from a foregone gain of same dollar amount. Therefore, buying the wrong asset and actually losing is often a more serious type of mistake than overlooking one which subsequently appreciates in value.

The vast majority of prediction models in the literature above assume a real-valued dependent variable and take symmetric loss around realized values. This approach works well for understanding market mechanisms, but ignores the intrinsic asymmetry when applied to investment decisions. Consider the following two prediction/realized value pairs: 1) flat zero prediction and prices goes up by 2% and 2) up 1% prediction and price goes down by 1%. Most existing loss functions (e.g. mean-squared error loss) treat these two prediction errors as equally serious – a 2% deviation from realized values, and resultingly so does the prediction model. However to the investor they are dramatically different: the first one amounts to a foregone serendipitous gain, the second one is a real loss. Missed opportunities are more easily accepted by resource constraint investors, but all try hard to avoid losses. To the best of our knowledge, most investment strategies use symmetric loss to generate predictions, and enter into a position if prediction magnitude is bigger than some threshold.

In this project, you are challenged to build a trading algorithm that achieves good common sense metrics in trading and appeals to investors' risk-averse psychology.

Data:

The raw data are the daily stock data of 50 companies, which are constituents of the SSE 50 index. The period of our data is from Day 1 to Day 756 (in the years 2017-2019). The data of one day includes open, high, low, volume, and adjusted close. The adjusted closing price, in particular, is the closing price that takes the dividends, stock splits, and other factors into account and hence is a better reflection of the current value of a stock. Thus, the adjusted closing price is used when calculating percentage change of the price of a stock and the return of a trade.

In the financial industry, practitioners would spend countless hours researching on new features in order to boost the prediction power of their learning algorithm. We have engineered some common features for you to use (see the `technical guidance`). You are also encouraged to engineer more predictors to achieve better performance. Note that, to simplify things for a class project, we did not consider the trade suspension issue. Given this simplification, `please do NOT engineer variables by pulling information across different stocks`.

Note also here that you will need to create the y (response) variable yourself when you formulate a prediction problem in the trading process.

Your mission:

On the evening of Day 504, you receive $\$1,000,000$ in your investment account which does not generate interest. From Day 505 to Day 756, you trade the 50 stocks described above or a subset of them. You should have cash only in the investment account by the end of Day 756. This means that you are required to sell off all your stocks on the last trading day. We take a `no short sale` rule (A short sale is the sale of an asset or stock the seller does not own) for our project which most large mutual funds also enforce. To simplify things, assume that you buy and sell each stock at the daily adjusted closing price. We take the $t + 1$ rule, which means if you buy a share of stock A at time t, you can only sell it at $t + 1$. We assume that you can purchase any fraction of a share. For instance, if a stock has price \$60 for a share and you decide to allocate \$90 to buy it, you can buy 1.5 shares. We also assume that whenever you decide to buy or sell a certain number of shares, you have a counterparty to engage in the transaction.

There is one thing that we do not simplify for this project: transaction cost. The transaction cost comes from two sources: stamp duty and slippage. Frequently, people in the financial industry adopt "basis point", or sometimes called"beep", as the unit, which is equivalent to 0.01%. The stamp duty is the tax charged by the government when you trade stock and we assume it is 10 beeps for a buy and sell roundtrip. The other cost, the slippage, refers to the difference between the expected execution price (the adjusted closing price in our case) of an order (either buy or sell) and the actual execution price. We take the value of 1.5 beeps for a single trade action, hence 3 beeps for a roundtrip. Together, for a roundtrip, the total transaction cost is 13 beeps, equivalent to 0.13%. So for a single trade action, the transaction cost is 6.5 beeps, equivalent to 0.065%. Take a concrete example for illustration. Suppose stock A's price is $1,000$ at time t. When you purchase one share at t, you will have to spend $1000 \times (1 + 0.065\%) = 1000.65$ to get it. Then suppose its share price rises up to $1,500$ at $t + 1$ and you want to sell your share. After the sale, you will get $1500 \times (1 - 0.065\%) = 1499.025$.

Suggestions and evaluation metrics:

Building a trading algorithm is more complicated than just running classification or regression on a given dataset. `First`, in addition to the given features, you might want to engineer new ones. `Second`, although most trading algorithms have regression in their backbone, there is recent evidence that with proper formulation, classification algorithms can also be the core. Hence, you might try both classification and regression techniques. `Third`, it would be too naive to just train one prediction model using data prior to Day 505 and use this model predict data from Day 505 to Day 756. You have much freedom in picking a proper training period and a prediction period. There is no unique answer to this. It is just one of many tunable components in your final trading strategy. `Fourth`, you should be careful in not using the future to predict the past. That said, your trading strategy at time t can only be built upon information up to $t-1$. `Fifth`, remember that after make predictions using your model, you still need to decide which stocks you buy or sell and how many shares would you like to buy or sell.

For the evaluation metrics, the `most obvious one` is the amount of money you have at the end of Day 756 after all the trading activities. The `second metric` is the `Sharpe ratio`, which considers both the return and the risk of a strategy. Usually, people prefer a higher Sharpe ratio. The preferred way to calculate the Sharpe ratio for our project is included in the `technical guidance`. A `third metric` is the number of days you make a profit in the trading period. This metric reflects some psychological effects of your trading algorithm.

This project includes:

1. `A final report` **due at 5 pm Los Angeles time on May 4th**, including

   i) Name and student ID of every member of the group on the cover page. Also indicate the contact person and his/her email.

   ii) Your understanding of the trading problem.

   iii) Description of the data.

   iv) Review of the statistical learning approaches that you tried and thought about trying.

   v) A clear description of the final trading strategy you used and why you chose it.

   vi) Summary of the results, including the total earning and sharpe ratio (whose definition can be found in the technical guidence), and the number of days your trading strategy incurs a loss.

   Note: In addition to the cover page, the final report should have **no more than 8 pages** (Font size must be 11 points or larger; reports exceeding the page limit will incur a penalty) and submitted in **pdf file** to the following link: https://www.dropbox.com/request/mxqlMFOiXQbv6MoCdaAY. Please name the file by the contact person's name. For example, if Mike Newton with USD ID 9382976532 is your contact person, the submitted file would be mike_newton_9382976532_project.pdf. `The instructor reserves the right to ask for your Python codes if he sees inconsistance or other problems in your pdf report`.

2. `A 10 min presentation` in class in **the second to the last week**.

   i) Summary of the problem. Your trading strategy and results

   ii) Graphics useful in communicating the results

   iii) All members of the group should appear in the zoom presentation. The presentation can be shared among a subset of the members, but all members should contribute to the preparation of this presentation. On the first slide, indicate the contact person and his /her email.

   iv) Q&A. The audience, including the instructor, can raise questions to any one of the group members

   Note: (1) In preparing the presentation, you should take the audience as a data-savvy manager who is smart with statistical training to the level of multiple linear regression but not beyond. (2) Due to the heavy imbalance of the session sizes, it is preferred that a group presents in the morning session if it has one group member from the morning session. A signup sheet will be created later.

Grading criteria:

The final project counts 13 out of 100 in the final grade calculation. All members of the same group will be assigned the same points. The points are decomposed as follows:

1. Presentation (6 points), based on time management, clarity, organization, statistical analysis, response to questions, etc. (0: a total mess; 1: poor; 2: fair; 3: good; 4: very good; 5: excellent; 6: exceptional)
2. Final report (7 points), based on writing quality, organization, statistical analysis, whether you have taken the constructive advices you receive during the presentation, etc. (0: a total mess; 1: poor; 2-3: fair; 4: good; 5: very good; 6: excellent; 7: exceptional)

Inevitably, the ratings will be somewhat subjective. But grades will be assigned in a consistent way.