LASSO Application in High-Frequency Trading

Zhengyuan Dong

Summary

The goal of the project is to explore the effect of Least Absolute Shrinkage and Selection Operator (LASSO) in the prediction of rolling 1-minute-ahead return. We are motivated by an awesome paper *Sparse Signals in the Cross-Section of Returns* (Alex, Adam, & Mao, 2017), which was published on the Journal of Finance. It is an explosive experiment for us, since some important details of how to implement the strategy are not disclosed in the paper. Our report mainly focuses on these details and how we really implement this algorithm to avoid similar content with the paper. We do get some meaningful results like average of 0.6% out-of-sample R^2 for the combination strategy of LASSO and AR(3) model. We mainly implement the project on Python and some SAS.

Motivation

Traditionally, researchers identity some candidate predictors like the unemployment rate, CPI, some indices, etc., measure their quality and build some models to predict future stock returns. But modern financial market is quite complex, it is hard to use intuition or some simple statistics to identify candidate predictors. Recently, some algorithmic traders predict stock price through sentiment analysis based on news, twitters, etc. LASSO is also a new way. The motivation of LASSO is simple. If we fit the linear regression by using all US stocks, there will be overfitting problems, since we get limited number of observations but too many predictors. LASSO is often used when the number of variables exceed the number of observations to avoid overfitting problem, which is exactly our case.

Data description

• Data Source

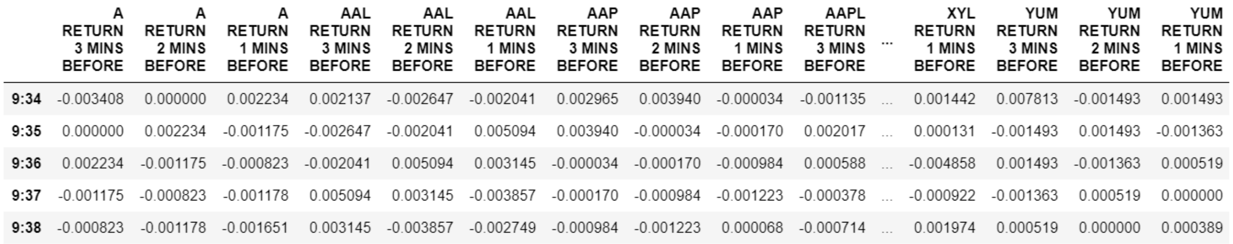
We get data from Trade and Quote (TAQ) of Wharton Research Data Services (WRDS) like the paper does. But the paper does not tell which part and how to use the data to generate returns. TAQ contains intraday transaction data for all securities listed on NYSE, AMEX, etc. We use the quote part to generate returns, because trading prices bounce and contain a lot noise, while quotes are relatively more stable. However, even for the same second, the data contains many bid and offer prices given by different exchanges. We use National Best Bid and Offer (NBBO), which is the lowest offer price and highest bid price among all exchanges at that time point, and we generate it through Cloud SAS Studio on WRDS. Then we get the mean of best bid and offer as the stock price. To get minute return, we use the last point of each minute as the price of that minute, and use log returns $r_t = \log\left(\frac{S_{t-1}}{S_t}\right)$.

• Data Structure

The paper originally uses 2000+ NYSE listed stocks, but out of computation consideration, we use SPDR ETF sector stocks, in total 474, by Apr. 2019. We use ETF sector stocks simply to achieve relatively more even distribution of stocks among different industries. Predictors are the lagged returns of each stock during the previous 3 minutes. Hence, for 1-minute-ahead return forecast of each stock, we get $474 * 3 = \ 1422$ predictors. For example, to estimate the return of AAPL at 12:04 $r_{AAPL,12:04}$, we use $\{\ r_{s_1,12:01},\ r_{s_1,12:02},\ r_{s_1,12:03},\ r_{s_2,12:01}, \dots,$ $r_{s_{474},12:01}, r_{s_{474},12:02}, r_{s_{474},12:03}$ | where S_i represents stock i , $i = 1, 2, \dots 474\}$ as candidate predictors. Intuitively, it is

$$r_{i,t} = \ \alpha_{i,t} + \sum_{j=1}^{474} \beta_{j,t-3} \cdot r_{j,t-3} + \sum_{j=1}^{474} \beta_{j,t-2} \cdot r_{j,t-2} + \sum_{j=1}^{474} \beta_{j,t-1} \cdot r_{j,t-1} + \varepsilon_{i,t}\ .$$

Since we use 3-min lagged returns, the first row is 9:34 for each trading day, and due to closing auction, we use 15:59 as the last row of observation. Hence, our data matrix (X) has 386 rows and 1422 columns which looks like below



	A RETURN 3 MINS BEFORE	A RETURN 2 MINS BEFORE	A RETURN 1 MINS BEFORE	AAL RETURN 3 MINS BEFORE	AAL RETURN 2 MINS BEFORE	AAL RETURN 1 MINS BEFORE	AAP RETURN 3 MINS BEFORE	AAP RETURN 2 MINS BEFORE	AAP RETURN 1 MINS BEFORE	AAPL RETURN 3 MINS BEFORE	...	XYL RETURN 1 MINS BEFORE	YUM RETURN 3 MINS BEFORE	YUM RETURN 2 MINS BEFORE	YUM RETURN 1 MINS BEFORE
9:34	-0.003408	0.000000	0.002234	0.002137	-0.002647	-0.002041	0.002965	0.003940	-0.000034	-0.001135	...	0.001442	0.007813	-0.001493	0.001493
9:35	0.000000	0.002234	-0.001175	-0.002647	-0.002041	0.005094	0.003940	-0.000034	-0.000170	0.002017	...	0.000131	-0.001493	0.001493	-0.001363
9:36	0.002234	-0.001175	-0.000823	-0.002041	0.005094	0.003145	-0.000034	-0.000170	-0.000984	0.000588	...	-0.004858	0.001493	-0.001363	0.000519
9:37	-0.001175	-0.000823	-0.001178	0.005094	0.003145	-0.003857	-0.000170	-0.000984	-0.001223	-0.000378	...	-0.000922	-0.001363	0.000519	0.000000
9:38	-0.000823	-0.001178	-0.001651	0.003145	-0.003857	-0.002749	-0.000984	-0.001223	0.000068	-0.000714	...	0.001974	0.000519	0.000000	0.000389

For each day, we have a new X matrix, and for each stock in the same day, X matrix remains the same, and Y is changing as stock changes.

To build models for each minute, we use previous 30 minutes as estimation window. For example, to predict the first point, we use the first 30 rows to build model, then apply the model to the 31st row and make the prediction, and for the next point, we roll one minute ahead.

Because we use 30-minute estimation window, the first prediction time is 10:04, the last prediction time is 15:59, and for each minute, we have a new model for each stock. Therefore, every day we have 356 models per stock.

Algorithm

• LASSO

LASSO is the simplest variable selection algorithm. It adds L1 norm to realize regularization on ordinary least square regression (OLS). It simply shrinks the coefficients of variables whose values are smaller than its penalty term λ to be zero. The formula of LASSO is

$$\min_{\beta \epsilon R^p} \left\{ \frac{1}{N} \left|\left|Y - X\beta\right|\right|_2^2 + \lambda ||\beta||_1 \right\},$$

where λ is the penalty term and is also the only parameter needs to be tuned. The larger λ is, the larger the penalty is, the fewer number of variables are selected. Reasonable value of λ for this case should select fewer than 30 variables since the number of observations is 30 for each model, if the number of variables exceeds 30, overfitting problems still exist. Fitting LASSO is quite time-consuming compared to fitting OLS or Ridge, since it does not have a direct solution.

• Cross Validation

It is the most important but meanwhile the most time-consuming part. We use 3-fold cross validation of the 30-minute estimation window to tune the parameter. For each minute, we do 3-fold CV to search for the optimal λ. It means for each minute, the program has to fit around 60 LASSO, because LassoCV in scikit-learn (Python library) uses grid search to get the optimal λ. To predict 50 stocks for a single day, the program has to fit around 50*60*356 = 1068000 LASSO regression, no wonder the program takes 4 to 5 hours to run for a single day.

The paper simply mentions it uses 10-fold CV and *glmnet* package in R to do it. But we wonder whether CV in the estimation by simply using package is correct. It is stock return, which means it is actually time series model, if we follow standard procedure of CV, using the model built on the last 20 minutes to fit and predict the first 10 minutes data seems quite unreasonable. Theoretically, rolling basis CV is more reasonable for time series data. But since the paper gets quite good results, we can ignore it.

• Fix λ for a period

In real life trading, it is more likely for traders to fix a model in the previous day and use it the next day, or at least fix the model for a period. It may not make sense to change the model every minute, that is, change λ every minute. So in addition to the paper, we also do CV on 1.5-hour window and apply the λ given by CV to the next 1.5 hours. One thing to note, though we fix $\boldsymbol{\lambda}$ for the next period, the coefficients of LASSO regression are still changing every minute because data changes. It significantly decreases the running time but also weakens the performance. For the same 6 days, this method only gives average 0.04% out-of-sample R^2. But we just randomly pick 1.5 hour as our period, we can trial more possible time lengths and select the best one with relatively good efficiency and performance.

• Out-of-Sample $\boldsymbol{R^2}$

We use out-of-sample R^2 to measure the return information explained by LASSO. It is a good metric to test the out-of-sample predictability of our model. Out-of-Sample adjusted R^2 is calculated

$$R_{oos}^2 = 1 - \frac{\sum_{i=1}^{n}(r_i - \widehat{r_i})^2 /(n-2)}{\sum_{i=1}^{n}(r_i - \overline{r_i})^2 /(n-1)}\,,$$

where r_i is the true return, $\widehat{r_i}$ is the predicted return, and $\overline{r_i}$ is the mean of true return. The paper fits the regression between true value and predictions, and then calculates R^2, which is the same as direct calculation via this formula.

• AR(3)

Since we incorporate 3-minute lagged returns of all stocks to make prediction, it is likely LASSO selects the stock's own 3-minute lagged returns as predictors, which is simply Autoregressive (AR) model, a classical time series model. Hence, to make sure LASSO explains extra information, we fit the regression between the true value and predictions given by AR(3) and LASSO, that is, $r_{true} \sim \widehat{r_{lasso}} + \widehat{r_{AR}}$, and get $R^2_{lasso,AR}$. Then we calculate partial R^2 by simply using $R^2_{lasso\,|\,AR} = R^2_{lasso,AR} - R^2_{AR}$, if $R^2_{lasso\,|\,AR} > 0$, we conclude LASSO explains extra information in addition to AR(3).

• Long-lived predictors

We also trial long-lived predictors to make predictions. We use iShare market ETF as market return, Russell 1000 as size return and Russell 2000 as value return as the paper did. We change the data matrix into 386 rows and 9 columns since we still use 3-min lagged returns. But this time, we don't need LASSO anymore because we only get 9 predictors. So we simply fit the linear regression and check the out-of-sample R^2, which is negative. Hence, long-lived predictors may not work well.

Limitation

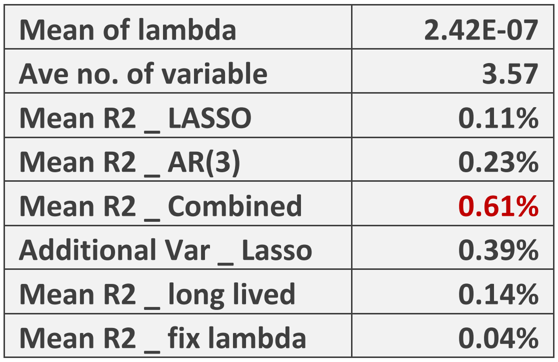
The main flaw of our project is that we do not use all NYSE listed stocks as candidate predictors. Because fitting LASSO for over 6000 predictors is much slower, we may need high performance computer to implement it. We trial Google Cloud Computing (GCP) which runs the program on cloud without occupation of our computers' CPU, but GCP does not decrease our running time, sometimes even increases to like 6 hours for a single day. Because of the computation and time limit, our results are not comparable to the results of paper no matter in the value of λ, number of selected variables or out-of-sample R^2.

Besides, we use 3-fold cross validation instead of 10 fold like the paper do. 10-fold theoretically will have better selection of the optimal λ , but doing so will triple our running time.

In addition, we only trial for 6 trading days (Dec 1, 2014 ~ Dec 8, 2014), and randomly select 50 stocks to make prediction. This will also limit the performance. If we refer to the *Average adjusted R^2 statistic each month plot* in the paper (page 14), we can see the fluctuation of R^2 for different months is large as the lowest to 1.4% and highest to 5%. Since we only generate data for December 2014, we can only get R^2 of this month. Longer prediction period and more number of stocks will improve the credibility of results.

Conclusion

Below is the table of summary statistics of this project.



Mean of lambda	2.42E-07
Ave no. of variable	3.57
Mean R2 _ LASSO	0.11%
Mean R2 _ AR(3)	0.23%
Mean R2 _ Combined	0.61%
Additional Var _ Lasso	0.39%
Mean R2 _ long lived	0.14%
Mean R2 _ fix lambda	0.04%

Mean of λ is 2.42e-07. The average number of variables selected by LASSO across 6 days' trial is 3.57, compared to average of 12 in the paper, it may be due to the fact that the author uses 2000+ NYSE stocks, while we use 474 stocks. LASSO alone does not perform well with only 0.11% information of stock return explained, which is rather inconsistent with the paper results. But the good thing is by combining with AR(3) model, the combined strategy explain 0.61% of information, which the highest one. Compared with using AR(3) alone, LASSO explains additional variation like 0.39%. We can see that fixing λ for longer time period doesn't have good result with only 0.04%. Though it is efficient, precision still needs to be improved.

In summary, we suggest using LASSO in addition to some preferred benchmark models like AR(3) or using long-lived predictors, which will produce better results instead of using LASSO or AR(3) alone. It is an explosive experiment for us. Due to computation limitation, we cannot produce the same level of R^2 like the paper, but we do learn a lot from the paper.

Reference

Alex, Adam, and Mao, 2017, Sparse Signals in the Cross-Section of Returns, *Journal of Finance*