Examining Efficient Market Hypothesis through Lens of Sentiment Analysis

Katya Chegaeva
Princeton University
Princeton, NJ 08544
chegaeva@princeton.edu

Xin Wan
Princeton University
Princeton, NJ 08544
xw13@princeton.edu

Abstract

The weak form of efficient market hypothesis (EMH) claims that stock prices contain all past price information. The semi-strong form says asset prices reflect all currently available information, including all news published anywhere in the world. We examine the validity of these claims by testing the predictability of Dow Jones Index (DJI) returns using news articles from Reddit.com. As an extension, we build trading algorithm for DJI based off of our predictive model. The performance of the predictive model as well as the trading algorithm is compared to various baseline models, including a naive model of predicting a constant, and a time series model developed by econometricians. Our results show that the semi-strong claim by EMH may not be valid with respect to the data and time period we chose to test on. In other words, our trading simulation shows one particular strategy could earn excess return with less market value volatility in the test period.

***K*eywords** Returns Prediction · Volatility Prediction · News Analysis · NLP

Motivation

Two forms of EMH is relevant to our study. The weak form, which is pioneered by Professor Malkiel at Princeton, states that no analysis of past stock returns could predict future returns. The semi-strong form, on the other hand, maintains that all public information - even non-financial ones contained in news articles - is reflected in stock prices. Econometricians have done extensive research on the weak form, but the semi-strong form has always been a challenge given the sheer amount of information contained in everyday news, as well as the complexity involved in extracting and quantify them.

Given the interconnection of economic, political, financial and social areas, we would expect an effect on news on financial sector not only be present for the new headlines on market activities but also on the general ones. From the "positive" sentiment on our bag of words of vocabulary we can see that it rapidly increases since 2008. Given the corresponding rise on the financial market in the same period we could assume at least some correlation present between these two variables. Our hypothesis that we can use sentiment in the news to predict the market moves can further be supported by the regression outputs we can see above for both volatility and return on the whole training set.

Variable	Coeff	P-value
Vol(-1)	0.1743	0.000***
Return(-1)	-0.0066	0.000***
'Positive'	-0.0008	0.085*
'Surprise'	0.0021	0.049**

Variable	Coeff	P-value
Vol(-1)	0.9990	0.045***
Return(-1)	-0.1047	0.000***
'Anger'	0.0314	0.067*
'Surprise'	-0.0436	0.068*

Regression resuls (only significant coefficients) on the whole dataset where we use sentiment vocabulary to calculate sentiment score for each day.

We can see the some interesting trends as well as expected relations from these regression outputs. Positive coefficient between for yesterday's volatility signifies the well-known fact of volatility clustering, while negative relationship between yesterday's return and today's volatility makes another economic rationale: lower return leads to higher volatility. Other than these phenomena, we can see that positive sentiment leads to the decrease in volatility and surprise results in the increase in one.

Returns regression also shows the increase in one as a result of previous period volatility, which can be attributed to the classical "high risk - high reward" relationship. Return also somehow negatively affects the next day's figure, indicating some kind of a negative autocorrelation pattern. We can also see some somewhat significant sentiment inputs into the return's explanation with anger affecting it positively and surprise - negatively.

All of the above give us a justification that sentiment analysis can allow us to better predict performance, hence, we implement the algorithm to do that.

Related Work

The possibility of making a fortune has attracted many onto the venture of predicting stock returns. Traditionally, econometric analysis - or "technical analysis" as dubbed on Wall Street - has focused on using past series of prices or returns to predict future ones. As early as in the 70s, Fischer Black and others have been studying the relationship between returns and volatilities. The famous article "Efficient Capital Markets" (1970) [1], however, dealt a blow to the aspiring traders in asserting that stock returns are in fact non-predictable – because news spread so quickly that they are reflected in prices with no delay. Therefore, prices always reflect all known information, including "predictable" events in the future. As a result, all subsequent price movements are by definition "a surprise". This school of thought has been dominant for a long time, and has given birth to the trillion-dollar industry of passive investment and ETFs, and financial service giants like Vanguard Group and BlackRock.

By the turn of the 21^{st} century, a new group of economists claiming to have found the secret recipe of predicting returns are again emerging. Among them, Lo and MacKinlay (1999) [2] reject the hypothesis that stock prices behave like true random walks by finding non-zero short-term serial correlation between stock movements. This is "momentum" in stock prices. Furthermore, behavioral finance researchers brought up the notion of "bandwagon effect", which explains momentum from a psychological perspective. Others focus on the long-term: Poterba and Summers (1988) [3] found that stock returns tend to mean-revert in the long run, and DeBondt and Thaler (1985) [4] attribute such behavior to investors overreacting to waves of optimism and pessimism. Despite some success in finding statistical significance, few has been able to identify profitable investment opportunities because transaction cost or tracking error usually wipe out any significant but small amount of excess returns. Moreover, so far the attention has been somewhat limited to the analysis of past movement of stock price per se, or maybe closely related economic variables like earnings or macro indicators. This will soon change.

As we march into the machine learning era, many novel approaches to combine data and models have been created, sentiment analysis being one of them. Kaur (2017) [5] develops an algorithmic trading mechanism using sentiment analysis and reinforcement learning which is able to achieve a Sharpe ratio of 2.4 given the baseline of -0.2. Gidofavli (2001) [6] also uses financial news article in order to classify the upcoming price series as "increased", "decreased" or "unchanged" using the interval of 40 minutes around the public information becoming available - or, namely, news release. Fung et. al (2005) [7] also investigate the market impact of news right after their release in accordance with Efficient Market Hypothesis. These analysis drastically expanded the scope of data included in the analysis, from periodic financial data to every news. The results are also encouraging. However, the claim of the (semi-strong) EMH is that, even if we manage to perfectly extract all information from all sources, we still should not be able to profit more than just holding a basket of stocks, without taking extra risk. This is what we are going to test on.

Data Description

The dataset comes from Kaggle.com, which includes both News and Dow Jones Index data from 08/08/2008 to 07/01/2016, with a total of 1,989 trading days in between. The news data comes from the WorldNews section on Reddit with a total of 73,608 Reddit headlines (on average, we have about 35 news articles per trading day). Each news article was top voted by users on that day. The Dow Jones Index represents 30 biggest US companies, including Apple, Boeing, IBM, Nike etc. Since these companies make up a large percentage of the total stock market, we use it as a proxy for the whole market. As these companies operate worldwide, it is reasonable to assume that the most important news on any day has some impact on the perceived future profitability of these firms, and hence their stock prices.

In order to work with a stationary time series data, we calculate log-returns from Dow Jones index. The return on date t is defined as:

$$r_t = \ln\left[\frac{P_t}{P_{t-1}}\right]$$

Furthermore, as a simplification (with empirical evidence) we assume mean of daily return is 0, and therefore the realized volatility on date t is:

$$RV_t = r_t^2$$

.

Another dataset we use is the sentiment vocabulary "NRC Word-Emotion Association Lexicon" (EmoLex) from the website of Saif M. Mohammad. The dataset consists of 10 sentiment scores on more than 14,000 words in English language. For each word and each sentiment, the score is 1 if the word is related to the sentiment category and 0 if not. For example, word "youth" has 'Positive', 'Anger', 'Anticipation', 'Fear', 'Joy' and 'Surprise' sentiments attributed to it, so it has score 1 in those categories, and 0 in others.

Methodology

Our analysis consists of four steps: feature extraction, model fitting and validation, prediction, and trading simulation.

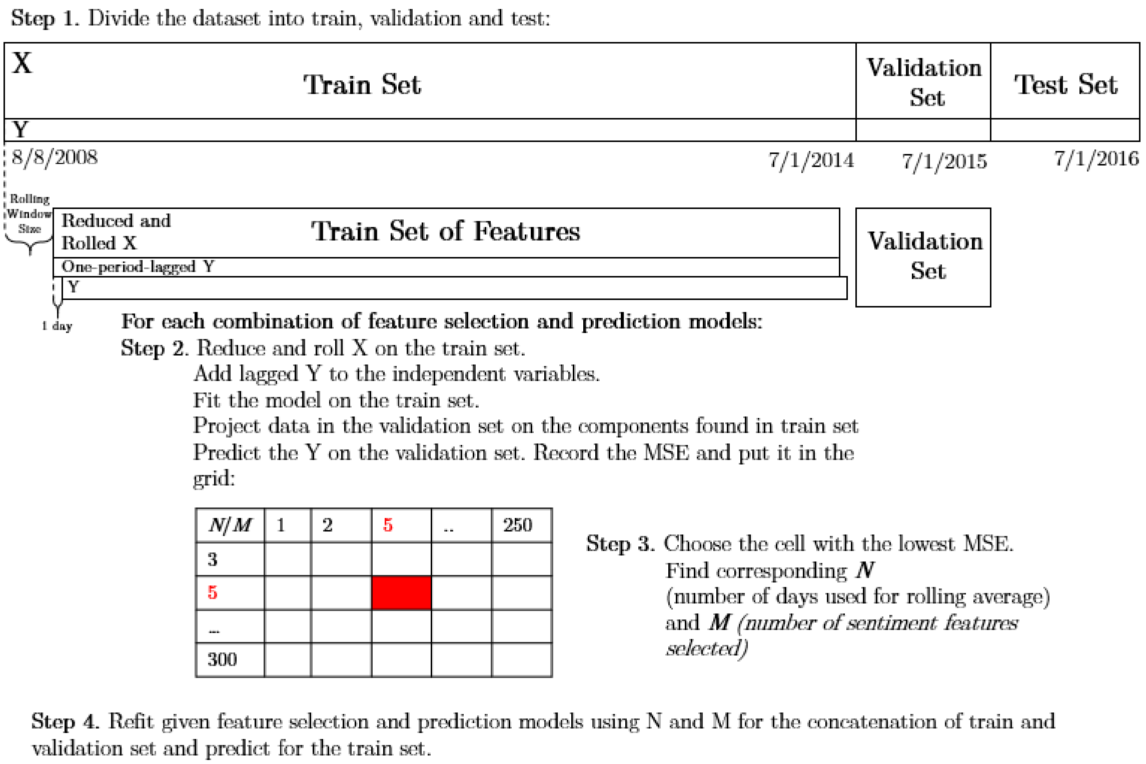
For the feature extraction part we use 4 different models, namely PCA, Factor Analysis, LDA and Sentiment scores. The first three models are standard unsupervised learning models we have covered this semester, while the last one is utilizing EmoLex. Specific processing steps are discussed below.

Before any feature selection model is applied, we first convert news article headlines to bag-of-words representation. We utilizes the NLTK library to tokenize, convert to lower case, and stem each word in the corpus of the news data. Then it extracts all distinct words (after previous processing) that appear at least 5 times to form the vocabulary. Each article is converted into a vector of integers with length equal to the length of the vocabulary. Hence the i_{th} entry in the j_{th} vector indicates the number of times the i_{th} word in the vocabulary appears in the j_{th} article.

For the first three methods we follow the same logic. Given our bag of words extracted from either Reddit or WSJ news headlines with a particular threshold of words, we calculate the average bag of words in a day by taking the average of all BOWs for that day. Furthermore, we conduct the feature extraction using PCA, Factor Analysis or LDA. For Sentiment Scoring we use the vocabulary described above where each word is scored across such sentiments as 'Positive', 'Negative', 'Anger', 'Anticipation', 'Disgust', 'Fear', 'Joy','Sadness', 'Surprise' and 'Trust'. Contrary to the unsupervised methods then features here are predetermined and we do not conduct any further feature selection. Each article will always be projected onto a 10-dimension space, each dimension contains the score of that particular sentiment in that article.

When it comes to the unsupervised methods (PCA, LDA, and FA), we need to select how many principal components / latent-variables / factors we are going to use. Another hyper parameter we need to tune is the size of the rolling window on which we average features. It comes from the assumption that our return or volatility tomorrow is not only affected by the sentiment in the news today, but also by a longer history. Hence, for example, using rolling window of the size 10 means that in order to predict the return tomorrow, we are going to use the average of the features of sentiments today and 9 days before today.

Hence, we need to choose the most appropriate combination of the size of the rolling window, and the number of features for a given feature selection and prediction model. In order to do so we want to explore which combination gives us the best possible performance on a held-out data: either the highest correct classification ratio for the returns' directions or the minimum mean squared error for the continuous predictions of volatility and returns. Hence, we explore numbers of components of: $[3, 5, 10, 20, 50, 100, 200, 300]$ and the rolling window sizes equal to: $[1, 2, 3, 5, 10, 20, 60, 120, 250]$. Based on all possible combinations of these values we construct a grid and, furthermore, choose the combination with the lowest MSE or highest CCR. However, we want to choose these parameters based on the held-out data, not on the train set, as train set would tend to favor higher number of features due to possible over-fitting. In order to do that we hold out one year of data from the train set right prior to our test set as a validation set. The algorithm is illustrated as in Figure 1.



Step 1. Divide the dataset into train, validation and test:

For each combination of feature selection and prediction models:

Step 2. Reduce and roll X on the train set.
Add lagged Y to the independent variables.
Fit the model on the train set.
Project data in the validation set on the components found in train set
Predict the Y on the validation set. Record the MSE and put it in the grid:

N/M	1	2	5	..	250
3					
5			🟥		
...					
300					

Step 3. Choose the cell with the lowest MSE.
Find corresponding ***N***
(number of days used for rolling average)
and ***M*** *(number of sentiment features selected)*

Step 4. Refit given feature selection and prediction models using N and M for the concatenation of train and validation set and predict for the train set.

Figure 1: Algorithm description to predict returns or volatility given a particular feature selection and prediction models.

Evaluation

Evaluating predictive power of our models

The goal of the analysis is to see if sentiment extracted from news data can improve predictive power of a model in addition to just using historical returns or volatilities. Therefore, comparing the performance of our model on normal data, versus the performance of the same model on a "control data" where the information from sentiment is taken out, can provide insights into whether the extra information is helpful.

We construct the "control data" by randomly shuffling news data across the whole data set. For example, news BOW vectors for date t_1 might ended up in the place for date t_2, and so on. The data for Dow Jones index remains order according to dates. Hence, we preserve the information related to historical price movements, but eliminate the information contained in news data because now news are no longer related to the return it tries to predict. We then run our fitted model on this new set and compute metrics such as MSE. The results are summarized below:

Trading simulation

We evaluate performances of our models by comparing them to various different baseline models. To evaluate the additional predictive power of sentiment on top of past prices, we create a new "control" data set by randomly shuffle the news data. If sentiment data actually improves prediction, our model predicting on actual data should outperform its prediction on the control data.

To put our model into the context of finance theory, we simulate several trading algorithms based off of our prediction model. In these simulations, we assume no transaction costs.

1. **Buy-and-hold**: this strategy simply invests a unit amount (one share) at the beginning of a period, hold the security till the end of the period, and then liquidate it. Hence, the cumulative PNL of this strategy will be exactly the same as the price movement of the stock we invest in.

2. **Unit long-short**: this strategy holds one share of stock at the end of date t if our model predicts the return on date $t + 1$ to be positive, and holds negative one share if our model predicts a negative return the next day.
3. **Sharpe ratio based long-short**: The Sharpe ratio is defined as "excess return per unit amount of risk". In a daily setting, risk-free return is basically zero, and expected Sharpe ratio will be simply expected return over expected volatility of the next day. The Sharpe ratio based long-short strategy will choose to hold a position proportional to the expected Sharpe ratio. For example, if the model predicts Sharpe ratio to be 1 for the next day, the algorithm will long 1 unit of the stock by end of day today. If the expected Sharpe ratio is negative, the algorithm shorts the amount equal to the absolute value of the expected Sharpe ratio.

Results

We analyze our results for returns and volatility looking at the MSEs for the test set. Our baseline MSE for the returns predictions is 0.000108679, which is the volatility of our actual returns on the test set. As it can be seen, we do not really achieve anything higher than for our models. This is explainable as returns, especially daily ones are typically very hard to predict. However, in order to investigate whether there is still some achieved advantage by using the sentiment, we compare our MSEs for the test set with the "Shuffled Dataset" MSEs. We can see that in value they are on par with each other, which may lead us to the conclusion that the sentiment doesn't bring any additional value. However, as our end result is the trading strategy we are yet to determine if our returns predictions are able to bring us significant profits.

MSEs for Returns Prediction

	OLS	Lasso	Ridge	Elastic Net
PCA	1.14E-04	1.09E-04	1.12E-04	1.09E-04
FA	1.10E-04	1.09E-04	1.10E-04	1.09E-04
LDA	1.12E-04	1.09E-04	1.13E-04	1.09E-04
Sentiment Score	1.12E-04	1.08E-04	1.09E-04	1.08E-04

MSEs for Returns Prediction with Shuffled Features

	OLS	Lasso	Ridge	Elastic Net
PCA	1.12E-04	1.09E-04	1.12E-04	1.09E-04
FA	1.10E-04	1.09E-04	1.10E-04	1.09E-04
LDA	1.10E-04	1.09E-04	1.10E-04	1.09E-04
Sentiment Score	1.12E-04	1.08E-04	1.08E-04	1.08E-04

Exploring the volatility MSEs, we compare it to the baseline figure of 4.98E-08, which is the squared mean of our actuals. We can see that contrary to the returns predictions, we are able to explain at least some variance in the next-day volatility by our features. However, this is not that surprising as on financial markets volatility is generally more sticky and, hence, easier to predict just using the historical information on it. In order to see if in case of volatilities we get any kinda of insight using sentiments, we look at the MSEs for the control shuffled dataset as well. As we can see that while volatility is still somewhat predictable, it probably comes not really from the news pieces of information, but rather from the autoregressive term we add to our regression.

MSEs for Volatility Prediction

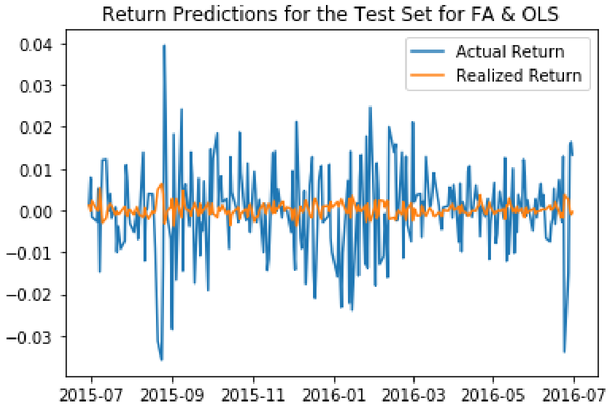
	OLS	Lasso	Ridge	Elastic Net
PCA	3.90E-08	3.86E-08	3.94E-08	3.86E-08
FA	3.58E-08	3.86E-08	3.58E-08	3.86E-08
LDA	3.61E-08	3.86E-08	3.57E-08	3.86E-08
Sentiment Score	3.82E-08	3.86E-08	3.87E-08	3.86E-08

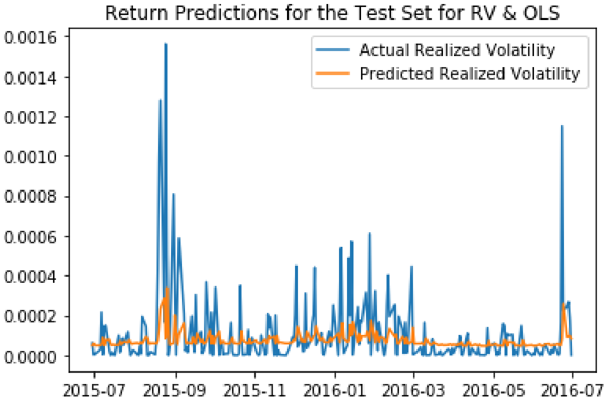
MSEs for Volatility Prediction with Shuffled Features

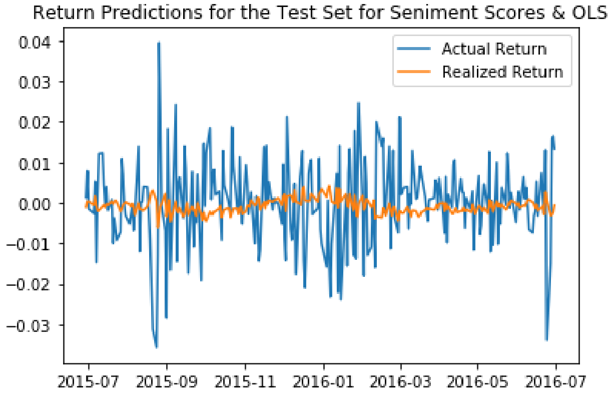
	OLS	Lasso	Ridge	Elastic Net
PCA	3.93E-08	3.86E-08	3.89E-08	3.86E-08
FA	3.58E-08	3.86E-08	3.58E-08	3.86E-08
LDA	3.60E-08	3.86E-08	3.57E-08	3.86E-08
Sentiment Score	3.82E-08	3.86E-08	3.88E-08	3.86E-08

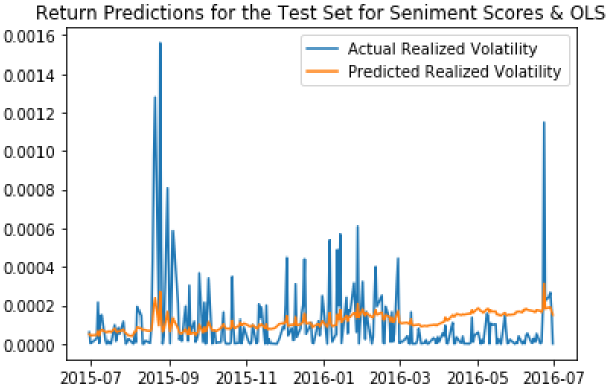
For both of the dependent variables we can tell that features selected in an unsupervised way do not significantly differ from the sentiment score ones in terms of a power of their predictions, which, despite being somewhat disappointing, makes sense, as if are not able to predict the next day's volatility and returns using sentiments, which we assume from the efficient market hypothesis, we would not be able to predict them no matter what "combinations" of the bag of words we use (which our underlying factors extracted from PCA or FA or sentiment scores calculated from the projection of our bag of words onto the vocabulary effectively are).

Another way we can see that returns and volatility generally have different predictive abilities due to their auto correlation structures (higher auto correlation for volatility and almost none for returns), we can look at the chart down below which reflect the predictions and actual values of our volatility and returns for the best model MSE wise with unsupervised feature selection and best sentiment score model.









As it can be seen once again, predictions for volatility are more precise than those for the returns, which explain almost zero variance, however, as we've investigated above it's more connected to the specifics of the time series other than the actual predictive nature of our sentiment.

Even though to further estimate the predictive power of our sentiment we are going to use our trading strategy results, we can still look at what different sizes of components and rolling windows were chosen in the volatility and returns modelling during our grid search.

Looking at the number of features chosen down below we can see that the number of them for OLS and Ridge is around 10 or 20. Lasso and Elastic Net choose much lower number of components. We attribute it to the fact that as these shrinkage models actually able to drop the features completely, they choose to do so because they actually shrink all features and only choose to leave 1 Intercept feature which we can also see from our MSE. The number of 10 or 20 for OLS and Ridge, however, seems reasonable as we would typically expect to be able to split news in around this number of categories. We could possibly think of such categories as "economic", "political", "positive", "negative", "world", "regional" etc.

Best Number of Features for Returns Prediction

	OLS	Lasso	Ridge	Elastic Net
PCA	20	3	10	3
FA	20	3	20	3
LDA	300	3	300	3

Best Number of Features for Returns Prediction

	OLS	Lasso	Ridge	Elastic Net
PCA	10	3	10	3
FA	3	3	3	3
LDA	50	3	50	3

Another result of our grid search is the size of the rolling window chosen by each model wither for volatility or return prediction. As we can see from down below, the number of days for the unsupervised feature selection for returns is always around 1 or 2, hence, assuming that return tomorrow is just explained by the news today and maybe yesterday. Looking at the sentiment score best rolling windows, however, we can see that they are slightly bigger which can be explained by a different nature of our features.

Best Rolling Window Size for Returns Prediction

	OLS	Lasso	Ridge	Elastic Net
PCA	1	1	1	1
FA	2	1	2	1
LDA	10	1	2	1
Sentiment Score	20	60	60	60

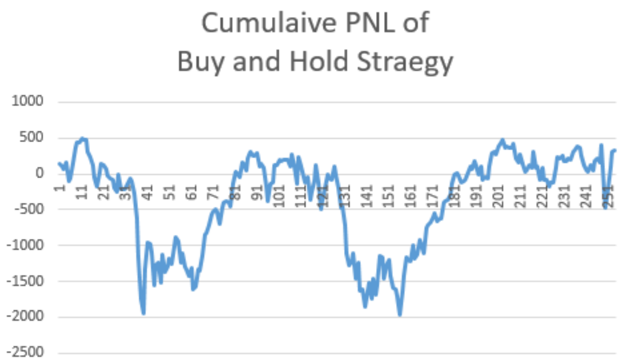
Best Rolling Window Size for Volatility Prediction

	OLS	Lasso	Ridge	Elastic Net
PCA	250	250	250	250
FA	250	250	250	250
LDA	250	250	250	250
Sentiment Score	250	250	250	250

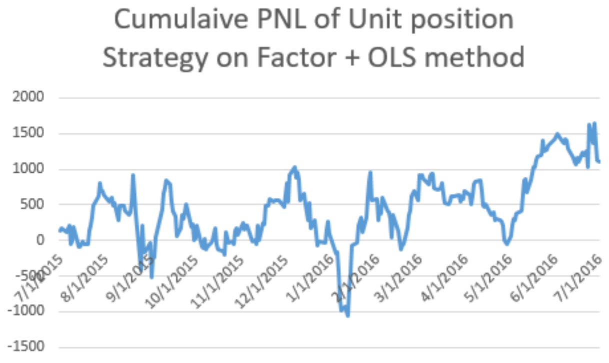
For the volatility though, both unsupervised selection models and sentiment ones agree that the volatility is a more long-term concept than return and you need a 250-day rolling window (which is basically one trading year) to better understand volatility tomorrow. This is explainable from both economic and financial perspective due to the fact that the volatility is a long-history dependent figure and also due to the notion of volatility clustering, stating that volatilities - high and low - seem to cluster with each other.

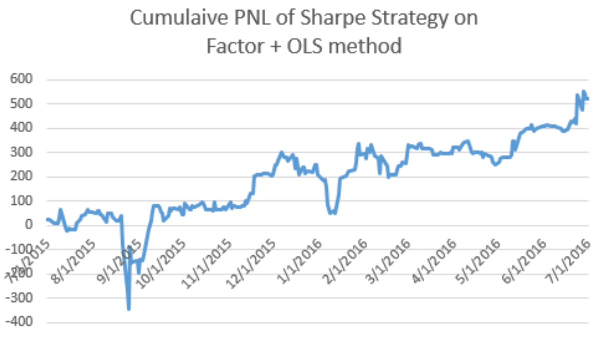
Trading Strategy Results and Comparison to Benchmarks

In order to see how our trading strategy performs in comparison to our benchmark, which is buy-and-hold, the profit for which is reflected below, we look at the cumulative P&L on our best MSE model (excluding Lasso and Elastic Net as they shrink the sentiment variables too much), which is Factor Analysis feature selection and OLS prediction model taking into consideration both return and volatility down below.



On the charts below we can that our profit based on our strategy is better than that for the buy-and-hold strategy. It's worth mentioning that even though the scales are different, we do not take them into consideration as much, due to the difference in position adjustment. While for the unit strategy, it's always 1 positive or negative position, for the Sharpe strategy the adjustment will be lower. What we care about is the general upward trend of our P&L and the ability to recover in periods of actual market drops.





Looking at the above, we can see that our strategy which incorporates sentiments actually performs well with a general upward trend in the portfolio position. We could take these results with a grain of salt and say that it performs better just due to the better volatility predictions. As in buy-and-hold strategy we just keep the constant amount of index, we do not really adjust our position at any point. However, as we actually build our strategy position based on volatility when it comes to the Sharpe ratio maximization strategy, it avoids getting new stocks at the periods of high volatility (which, arguably, from our Buy-and-Hold strategy graph, we are in during our test set). It would also explain why our Sharpe model performs better than the unit model, which doesn't take the volatility into account, where by better we mean a more constant upward growth rather than a more volatile one in the unit strategy, where our position fluctuates from positive to negative for a majority of period.

However, we could also claim that some of that is still attributed to the news. While MSEs for returns calculated above indeed do not seem to show that our predicted returns explain any volatility in our actual returns, we could argue that while for the whole period it could be the case, it may not actually be the case where Dow Jones index fell (which is the drop on the buy-and-hold strategy chart). As it can be seen, other than two significant drop periods, Dow Jones index seemed to rise. Hence, if our model helped foresee those drops and avoid them, it would explain why our strategy performs better even despite relatively high MSEs.

Conclusion and Future Work

In the present project we tried to see whether news sentiments help us improve our financial market performance. We found that even though our prediction results on returns do not explain the variance of our actual returns, our Sharpe ratio strategy outperforms buy-and-hold one. We could explain it with the volatility adjustments and, hence, not holding as much of index during the times of distress as we would under the buy-and-hold strategy. However, we claim that sentiment could also contribute to this result even despite low predictability power for the whole year. While the whole test set prediction performance is not really good, we still see that in the times of index drops our strategy doesn't lead to the drop of our position, hence, meaning that at those particular times, our sentiment somewhat helped us to predict the drop and avoid that. This, on one hand, rejects the semi-strong form of the efficient market hypothesis as lets us see that we can make returns out of public information available on the market.

This project could be further developed not only to look at news' headlines, but also at the body of the news itself, as we can imagine that headlines represent only a limited information on the event and can often be exaggerated to attract attention of the readers. Furthermore, we could significantly improve the quality of our insights by using the intraday data. For that we would also need to have a particular timestamp on each piece of news, but that would allow us to much better track the origin of effects arising.

References

[1] Burton G. Malkiel and Eugene F. Fama. Efficient capital markets: A review of theory and empirical work*. *The Journal of Finance*, 25(2):383–417, 1970.

[2] Andrew W. Lo and A. Craig MacKinlay. *A Non-Random Walk Down Wall Street*. Princeton University Press, 1999.

[3] James M. Poterba and Lawrence H. Summers. Mean reversion in stock prices: Evidence and implications. *Journal of Financial Economics*, 22(1):27 – 59, 1988.

[4] Werner F. M. DeBondt and Richard Thaler. Does the stock market overreact? 1985.

[5] Simerjot Kaur. Algorithmic trading using sentiment analysis and reinforcement learning. 2018.

[6] Gyozo Gidofalvi and Győző Gidófalvi. Using news articles to predict stock price movements, 2001.

[7] Hongjun Lu Gabriel Pui Cheong Fung, Jeffrey Xu Yu. The predicting power of textual information on financial markets. 2005.