Machine Learning, Investor Sentiment and the Financial Market

Iles Acly Ayati

September 19, 2019

Introduction

This thesis concerns akdjskadlskdfosf In the recent years, machine learning tools may seem to many like buzzwords, especially in finance, where few understand what they actually can and cannot produce. They are continuously changing the playing field and are increasingly deployed in analytic departments across the industries. However, there are limitations to all tools, and the issues associated with financial time series predictions still prevail - the time series are inherently noisy, complex and chaotic (Kumar and Thenmozhi, 2006; Kara et al., 2011; Patel et al., 2015). But that has not stopped researchers from adopting new and experimental techniques in their pursuit. An example of innovative adaptations of machine learning tools is Natural Language Processing. It is used to analyze the effects of web texts for stock market predictions (Das and Chen, 2007; Tetlock, 2007; Tetlock et al., 2008). Words from web articles are collected and transformed into measures of investor sentiment by their positive or negative charges, before they are fed to neural networks along with stock returns. Findings show that, with varying accuracy, market events as reported in financial news articles can indeed be transformed into sentiment measures and used to predict prices. Another example is Patel et al. (2015), which compares four popular ensemble algorithms; Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest and Naive Bayes. The study focuses on data pre-processing where continuous data is transformed into discrete data to increase prediction performances. The results show that Random Forest produces the best classification predictions. Tsai et al.

(2011) also applies classifier ensemble methods to analyze stock returns. It compares prediction performances using hybrid methods of majority voting and bagging, as well as combining two types of classifier ensembles with single classifiers. The findings are somewhat conflicting, but they do conclude that on average, combining multiple classifiers (e.g. an ensemble of ANNs, random forest classifiers, and logistic regressions) provide better prediction accuracies than a homogenous ensemble of classifiers (e.g. ensemble of ANNs or random forest classifiers only).

Related Literature

Empirical studies on stock price predictions are among the most popular in financial market research. Economists are divided when it comes to modelling price movements. On one side, you have the proponents of randomness in the prices movements - they don't follow any measurable patterns, and are thus impossible to fully predict. Samuelson (1965) set to prove that stock prices follow a random walk if rational competitive investors require a fixed rate of return. *Efficient capital markets: A review of theory and empirical work* (Malkiel and Fama, 1970) is arguably the most cited paper in the history in financial research after its demonstration that stock prices are indeed close to a random walk.

The opposing view that market movements are rooted in human behavior and our expectations - which are not always rational, has also been well documented, though usually on a more anecdotal basis. For instance, the early 60's were characterized by a high demand for small, newly issued growth stocks (Malkiel, 1999). In pages 55-57, Malkiel mentions a 'new-issue mania' that later led to a decline in growth stocks much stronger than the rest of the market. Investors seemed to have ignored the uncertainty of newly issued electronics stocks in favor of promised overnight returns. On the other hand, De Long et al. (1990) takes a more theoretical approach, and argues that changes in investor sentiment are not fully countered by arbitrageurs and therefore affect security returns. Arbitrageurs are "fully rational" investors who trade to exploit that the price of a security may sometimes deviate from that of its perfect substitute - a portfolio of other securities with the same risk and return profile. For example, if the price of a security is below the price of the substitute portfolio, arbitrageurs will sell the portfolio and buy the security until the prices are back to equal. When the substitute is indeed perfect, this arbitrage is completely riskless. Thus, arbitrageurs have perfectly elastic demand for the security at the price of its substitute portfolio. Such riskless arbitrage is very effective for derivative securities such as futures and options, but for individual stocks and bonds, perfect substitutes are usually not available. Instead, *close* substitutes are traded, which implies that the arbitrageurs' demand is no longer perfectly elastic as there will exist some degree of risk-return differences between the security and its close substitute. In that regard, De Long et al. (1990) argues that arbitrageurs will not be able to completely offset investor sentiment. This "noise trader approach" was further developed upon by other economists. Lee et al. (1991) examined closed-end funds and proposed that their discount fluctuations are driven by investor sentiment. One thing that both schools of thought agree upon is the difficulty in predicting asset price movements. Many studies aim to provide a rational model directly proving that prices are indeed predictable (De Bondt and Thaler, 1985; Jegadeesh, 1990; Lo and MacKinlay, 1990). However, profiled economists seem to share the opinion that they don't really add more to what we already have in form of the Capital Asset Pricing Model (CAPM). [1]

The Latin statement *'Cum hoc ergo propter hoc'* is a reminder about the logical fallacy we are faced with when drawing conclusions that are exclusively based on observations of two events occurring simultaneously. In English, this is known as 'correlation is not causation'. Granger (1969) neatly provided the framework to test for causality between time series, and is widely recognized as *the* method for investigating causal relationships in econometrics. The basic premise of the paper relied heavily on the notion that cause cannot happen after effect. Consequently, it focused on 'lagged' causality in linear relationships. Granger's definition of causality is as follows: X_t is causing Y_t if we are better able to predict Y_t using all available information U than if the information apart from X_t had been used. Obviously, using 'all available information' is impossible, which in practice means that we instead have to select the relevant information for the time series Y_t. To test for instantaneous causality, we would also include the contemporaneous or new information in the universe U, while lagged causality will only contain information up to and including $t-1$. Due to its simplicity it gained a lot of popularity as well as criticism by economists. The paper was later extended to include frameworks for instantaneous causality as well as tackling some issues and limitations regarding non-linear and deterministic relationships. In an efficient market, the best price forecast of

[1] Thaler, R.H. and Fama, E. F. (2014, October 18). Chicago Booth Review Interview: https://review.chicagobooth.edu/economics/2016/video/are-markets-efficient

tomorrow's prices are today's prices, since the probability of an increase equals that of a decrease (Malkiel and Fama, 1970). By that reasoning, the presence of causality could be interpreted as evidence of market inefficiencies of the weak form (Niarchos and Alexakis, 1998). However, not all economists agree on this point. Two cointegrated time series as defined in Engle and Granger (1987) implies Granger causality in at least one direction, but has been shown that there is no general equivalence between the existence of arbitrage opportunities and cointegration (Dwyer Jr and Wallace, 1992). In finance, most studies on causality focus on the relationship between stock prices and their respective trading volume, where most test for instantaneous causality (Hiemstra and Jones, 1994). However, there are studies in this field relying exclusively on traditional, lagged Granger causality tests (Smirlock and Starks, 1988; Jain and Joh, 1988). Although such tests can be good in explaining linear causal relations, their usefulness in non-linear casual relations is debatable (Hiemstra and Jones, 1994). For this reason, traditional Granger causality tests might overlook significant non-linear patterns that machine learning algorithms may capture.

In the recent years, machine learning tools may seem to many like buzzwords, especially in finance, where few understand what they actually can and cannot produce. They are continuously changing the playing field and are increasingly deployed in analytic departments across the industries. However, there are limitations to all tools, and the issues associated with financial time series predictions still prevail - the time series are inherently noisy, complex and chaotic (Kumar and Thenmozhi, 2006; Kara et al., 2011; Patel et al., 2015). But that has not stopped researchers from adopting new and experimental techniques in their pursuit. An example of innovative adaptations of machine learning tools is Natural Language Processing. It is used to analyze the effects of web texts for stock market predictions (Das and Chen, 2007; Tetlock, 2007; Tetlock et al., 2008). Words from web articles are collected and transformed into measures of investor sentiment by their positive or negative charges, before they are fed to neural networks along with stock returns. Findings show that, with varying accuracy, market events as reported in financial news articles can indeed be transformed into sentiment measures and used to predict prices. Another example is Patel et al. (2015), which compares four popular ensemble algorithms; Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest and Naive Bayes. The study focuses on data pre-processing where continuous data is transformed into discrete data to increase prediction performances. The results show that Random Forest produces the best classification predictions. Tsai et al. (2011) also applies classifier ensemble methods to analyze stock returns. It compares prediction performances using hybrid methods of majority voting and bagging, as well as combining two types of classifier ensembles with single classifiers. The findings are somewhat conflicting, but they do conclude that on average, combining multiple classifiers (e.g. an ensemble of ANNs, random forest classifiers, and logistic regressions) provide better prediction accuracies than a homogenous ensemble of classifiers (e.g. ensemble of ANNs or random forest classifiers only).

Data and Feature Definitions

The collected data sets consist of monthly figures from January 1994 to December 2018. The number of observations varies between 279 to 299, depending on the model and the number of lags used. When no lags are used, all 299 observations are included. All variables are percentage changes. Response variables are already prepared in the Kenneth French Data Library as monthly percentage changes, so no further manipulations are made. The sentiment measures, however, are a mix of index prices and level figures. We compute the log change of all the features' time series. This is to avoid any issues with percentage changes of negative numbers, as most of the features in the Commitment of Traders (COT) data set are net levels that fluctuate on both sides of zero. We proceed to more comprehensive definitions of each variable below.

Response Variables

Following the logic of Lee et al. (1991) we expect that the closed-end fund discount is better at explaining the variance in small stock returns than big stock returns. On the other hand, one could hypothesize that due the aggregate nature of the COT features, they are better at explaining big stock returns. The same logic applies to the CBOE Volatility Index, which is directly derived from the total S&P 500 and not its subsets. Therefore, all regression routines will be performed on a range of stock portfolios to better capture their performance on different subsets of the S&P 500.

The Kenneth French Data Library (KFDL) contains many portfolios formed on sets of characteristics such as size, operating income, investment levels etc. The chosen response variables in this paper are *6 Portfolios Formed on Size and Book-to-Market* from KFDL. These portfolios are constructed at the end of each June. They are the intersections of two portfolios formed on size (size of market equity, now ME) and three portfolios formed on the ratio of book equity to market equity (BE/ME). The size breakpoint for year t is the median NYSE market equity at the end of June of year t. BE/ME for June of year t is the book equity for the last fiscal year end in t-1 divided by ME for December of t-1. The BE/ME breakpoints are the 30th and 70th NYSE percentiles.

Sentiment Features

Closed-End Fund Discount: The difference between a publicly traded fund's price and the net asset value of this fund. This should in theory be close to zero, yet because these funds are traded only secondhand, meaning no new issues or buybacks. The supply of shares is constant, while the demand for these funds is subject to investor sentiment which leads to price-to-nav deviations.

$$cefd_t = \frac{NAV_t - \mathit{Trading\ Price}_t}{NAV_t}, \quad where \quad NAV_t = \frac{\mathit{Assets}_t - \mathit{Liabilities}_t}{\mathit{Number\ of\ outstanding\ shares}_t}.$$

CBOE Volatility Index: A market index that represents the market's expectation of 30-day forward-looking volatility. It is derived from the price inputs of the S&P 500 index options and provides a measure of market risk and investors' sentiment. Implied volatility is calculated by taking the market price of an option, entering it into the Black-Scholes formula, and back-solving for the value of the volatility.

$$vixret_t = \log\left(\frac{VIX_t}{VIX_{t-1}}\right).$$

Historical CFTC Commitment of Traders Report for Financial Futures:
The COT reports provide a breakdown of each Tuesday's open interest for futures and options on futures markets in which 20 or more traders hold positions equal to or above the reporting levels established by the Commodity Futures Trading Commission (CFTC).

The COT reports are based on position data supplied by reporting firms (e.g. FCMs, clearing members, foreign brokers and exchanges). While the position data is supplied by reporting firms, the actual trader category or classification is based on the predominant business purpose self-reported by traders on the CFTC Form 401 and is subject to review by CFTC staff for reasonableness. CFTC then aggregates the data and publishes it every Thursday. Thus, all the figures are of aggregate nature, which simplifies matters for researchers.

OI: Open Interest is the total of all S&P500 futures contracts entered into and not yet offset by a transaction, delivery, or by exercise. The aggregate of all long open interest is equal to the aggregate of all short open interest. Here, we use the log-change in Open Interest as an investor sentiment measure.

$$OI_t = \log\left(\frac{\mathit{Open\ Interest}_t}{\mathit{Open\ Interest}_{t-1}}\right).$$

CPNL: Commercials' Pressure Net Long. Here defined as the log-change in Commercials' long-to-short ratio (LS) on S&P500 futures. Commercial traders are defined by the CFTC as producers, merchants, processors and users of the physical commodity who manage their business risks with use of futures or option markets.

$$CPNL_t = \log\left(\frac{\mathit{Commercial\ LS}_t}{\mathit{Commercial\ LS}_{t-1}}\right).$$

NCPNL: Non-Commercials' Pressure Net Long. Here defined as the log-change in non-commercials' long-to-short ratio on S&P500 futures. Non-commercial traders are defined by the CFTC as professional money managers (e.g. CTAs, CPOs, and hedge funds) as well as a wide array of other Non-commercial (speculative) traders.

$$NCPNL_t = \log\left(\frac{\mathit{Non\text{-}commercial\ LS}_t}{\mathit{Non\text{-}commercial\ LS}_{t-1}}\right).$$

TOTPNL: Total Reportables' Pressure Net Long. Here defined as the log-change in total reportables' long-to-short ratio on S&P500 futures: Total Reportables refers to the sum of futures contracts held by commercial and non-commercial traders registered by CFTC.

$$TOTPNL_t = \log\left(\frac{\textit{Total reportables } LS_t}{\textit{Total reportables } LS_{t-1}}\right).$$

NONPNL: Non-Reportables' Pressure Net Long. Here defined as the log-change in non-reportables' long-to-short ratio on S&P500 futures: The sum of non-reportable contracts should always offset the sum total reportables. Because of this duality, only NONPNL is used in the regressions.

$$NONPNL_t = \log\left(\frac{\textit{Non-reportables } LS_t}{\textit{Non-reportables } LS_{t-1}}\right).$$

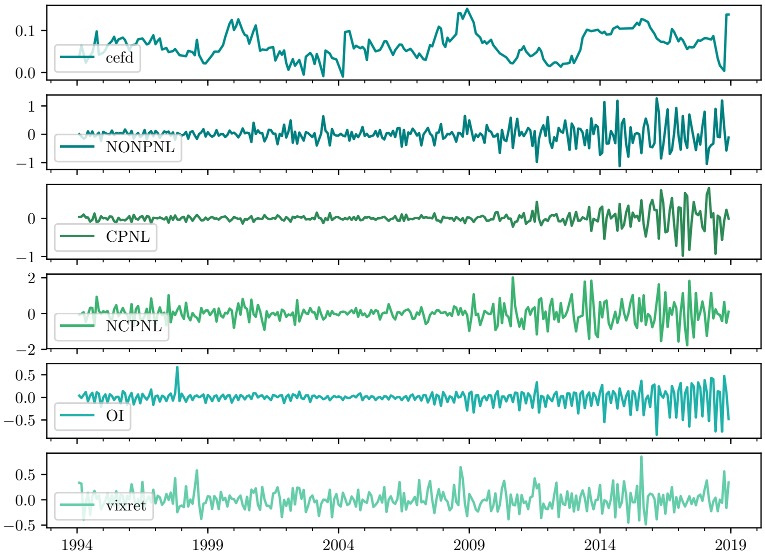
Table 1: First five observations of the chosen features

	cefd	NONPNL	CPNL	NCPNL	OI	vixret
1994-02-28	1.385	0.004	0.044	-0.014	0.038	0.399
1994-03-31	-0.257	-0.070	0.053	-0.005	-0.013	0.375
1994-04-30	0.231	-0.133	0.134	0.016	0.051	-0.327
1994-05-31	-0.676	-0.081	0.031	-0.086	0.120	-0.054
1994-06-30	0.285	0.079	0.033	-0.012	-0.090	0.149

Table 2: First five observations of the responses

	SMALLLoBM	ME1BM2	SMALLHiBM	BIGLoBM	ME2BM2	BIGHiBM
1994-02-28	-0.009	0.006	-0.011	-0.018	-0.035	-0.049
1994-03-31	-0.063	-0.050	-0.044	-0.049	-0.040	-0.042
1994-04-30	-0.008	0.006	0.013	0.003	0.019	0.013
1994-05-31	-0.026	-0.012	-0.001	0.013	0.013	-0.011
1994-06-30	-0.048	-0.022	-0.022	-0.033	-0.023	-0.026

Figure 1: Time series of features



Methodology

In this thesis we want to investigate the relationship between a selected basket of investor sentiment measures and a small range of stock market portfolios. More specifically, we want to test the hypotheses that these features may have explanatory power as well as test whether there exists a causal relationship between the features and portfolios. Moreover, we want to test *what* feature is the best in doing this.

The question regarding investors' rationality is, as previously mentioned, a hard one to answer. For instance, Warren Buffet's opinion is that the U.S. stock market is indeed rational most of the time, but he acknowledges that it has its occasional 'seizures'. Investor sentiment is often confused with the irrationality usually associated with those periods of 'seizure' in the sense that it is hard to measure it and it is deeply rooted in human behavior. Consequentially, one may think that sentiment has an insignificant role in explaining price movements over time, and that it should certainly not be able to forecast them. However, recall that an efficient market implies that prices reflect ALL current and past relevant information, which would include investor sentiment as well. Just because it is hard to measure does not mean it no longer qualifies as information that prices reflect. For this reason, it is important that we clearly define what we are actually testing. We are NOT testing the inefficiency of the market. Instead, we are simply looking for evidence that investor sentiment can both explain and cause price movements.

To do this, we first have to clear some definitions. The term 'prediction' is used a lot in statistical inference, but it does not have a universal definition. For this thesis, the term prediction will be a general reference to any predictions made from any model. 'Forecasts', on the other hand, will be used when referring to predictions made from lagged time series regressors exclusively. For example, a prediction from a vector autoregressive model would fall under the term 'forecast', since it uses lags of one time series to predict another, but a contemporaneous prediction from an ordinary least square model would not. This distinction is important to avoid any confusion when the models are compared.

The approach can be divided into three folds or overlapping phases where:
Phase 1) uses generalized least square to test the features' and their principal components'

explanatory power. Then, Phase 2 takes vector autoregression and ensemble algorithms to test for causality. Finally, Phase 3 uses singular values decomposition and ensemble algorithms for feature selection.

Univariate Generalized Least Squares

First, we want to analyze how the features perform individually. To do this, we first perform a sequence of univariate Ordinary Least Squares (OLS) regressions for each feature on each of the six portfolios. By doing this we also obtain information about the covariance structure of the residuals. Below, the portfolios are for simplicity called $response_i$, where $i = \{1, \cdots, 6\}$. Since we are repeating the regressions of all six features on all six portfolios, a total of 36 regressions are made.

$$\begin{aligned}
&response_{it} = \alpha_i + \beta_1 cefd_t + \epsilon_{it} \quad ,\\
&response_{it} = \alpha_i + \beta_1 NONPNL_t + \epsilon_{it} \quad ,\\
&\qquad\qquad\vdots\\
&response_{it} = \alpha_i + \beta_1 vixret_t + \epsilon_{it} \quad .
\end{aligned} \tag{1}$$

In traditional OLS regressions we use the observed data $\{y_t, x_{i,t}\}_{i=1,\ldots,n,t=1,\ldots,T}$ and minimize the sum of squared residuals of $\quad Y = X\beta + \varepsilon.$ Then, so long as the Classical Linear Regression Model (CLRM) assumptions hold, $\quad \mathrm{E}[\varepsilon \mid X] = 0, \quad$ and $\quad \mathrm{Cov}[\varepsilon \mid X] = \Omega = \sigma^2 I_n$. An important note is that the constant variance assumption of CLRM implies that the covariance matrix Ω is diagonal and constant $\sigma^2 I$. The estimated coefficients are then be unbiased, consistent and efficient. However, taking a quick glance at the features over time, some of them seem to have time-varying volatility. which can insinuate heteroscedastic variance in the residuals. While the coefficients will remain unbiased and consistent (i.e. the estimated coefficients reflect the true coefficients), they will not be efficient - their standard errors will likely be underestimated. Recall that the main reason for doing these univariate regressions is to paint a realistic picture of the explanatory power of these features. A quick glance at Figure 1 gives reason to suspect heteroscedastic errors, and hence, we perform a White's test to confirm whether this is true or not.

We proceed to use a Generalized Least Squares (GLS) model to better capture the true significance of each feature. The optimization problem is:

$$\hat{\beta} = \underset{b}{\operatorname{argmin}}\,(Y - Xb)^{\mathsf{T}}\,\Omega^{-1}(Y - Xb)$$

The procedure begins by factoring the non-diagonal covariance matrix Ω such that $\Omega = CC^{\mathsf{T}}$, where $C = \Omega^{-\frac{1}{2}}$. Then the transformation $C^{-1}Y = C^{-1}X\beta + C^{-1}\varepsilon$ will assure constant variance in the residuals. If we let $\varepsilon^* = C^{-1}\varepsilon$ then it can be shown that $\operatorname{Var}[\varepsilon^* \mid X] = C^{-1}\Omega\left(C^{-1}\right)^{\mathsf{T}} = I$. The coefficients will remain the same, but will have slightly different standard errors. They are estimated by $\hat{\beta} = \left(X^{\mathsf{T}}\Omega^{-1}X\right)^{-1}X^{\mathsf{T}}\Omega^{-1}Y$.

This shows that a GLS regression is equivalent to performing an OLS regression on a linearly transformed version of the data, which means that the interpretation of the results will remain the same.

Multivariate Generalized Least Squares

To examine how the features perform together, we proceed with multivariate GLS regressions. The categorization of the feature-groups is based on the source of the data; The COT features represent the first group, and the second group is all of the features bundled together. However, this poses an issue: The features are correlated, which means that CLRM's linear independence assumption $\Pr\big[\operatorname{rank}(X) = k_{\text{features}}\big] = 1$ is violated. Measuring joint significance is primarily for comparison with the machine learning approach later and the comparison will not make a lot of sense if we have unrealistic results in this part. There are a few solutions to the multicollinearity issue shown in Table **??**, but most of them involve making changes to the data, i.e. either cut out some features or changing the number of observations. Cutting out features would defeat the purpose of our feature selection goal, and increasing the number of observations is not feasible as collecting more data would require new channels and permissions that are not available. Thus, we can either ignore it, use a subset of the data, or transform the feature vectors so that they are orthogonal or linearly independent. Using a subset of the time series to estimate the coefficients and then apply them to the whole series is one solution. Another one is extracting their principal components and use them as features. Doing that will retain most of the information of each feature and simultaneously assure linear independence. Since the primary purpose of these multivariate regressions is to compare with the machine learning approach later, where we will use principal components anyway, we use the principal components here as well.

We then perform another GLS regression routine on all $i = \{1, \cdots, 6\}$ portfolios and take note of the group-wise significance of the features. Note that this test routine consists of 12 GLS regressions:

$$response_{it} = \alpha_i + \beta_1 NONPNL_t + \beta_2 CPNL_t + \beta_3 NCPNL_t + \beta_4 OI_t + \epsilon_{it} \quad (2)$$

$$response_{it} = \alpha_i + \beta_1 cefd_t + \beta_2 NONPNL_t + \beta_3 CPNL_t + \beta_4 NCPNL_t + \beta_5 OI_t + \beta_6 vixret_t + \epsilon_{it}$$

Principal Components Analysis

Principal Component Analysis (PCA) is a method that is often used to reduce the dimensionality of large data sets. It transforms a set of variables into a smaller one such that it still contains most of the information of the large set. In python, both the Statsmodels module and the Sci-kit Learn (SKlearn) library can be used to perform PCA. For the multivariate regressions, using the statsmodels.PCA function is straightforward and easy: We simply set it to standardize the variables and use the Singular Value Decomposition (SVD) method to obtain the principal components. The SKlearn approach is slightly different, though only because of the data splitting. The formal procedure stays the same: First, the data is standardized such that all vectors have a zero mean and standard deviation equal to one. This is done to assure that the relative scaling of the vectors is uniform across all variables so no variables weigh more than others when the principal components are obtained. Then, SVD is applied to transform the scaled data into principal components and their weights. The mathematical representation of the SVD procedure is as follows:

Let $\Sigma_{n \times p}$ be a diagonal matrix of positive numbers σ_k called the singular values of the feature matrix X. Singular values are equal to the square root of the eigenvalues λ_k of $X^{\mathsf{T}}X$. Now, let $U_{n \times n}$ and $W_{p \times p}$ be matrices where all columns are orthogonal vectors. The columns of U are called the left singular vectors of X, while the columns of W are the right singular vectors. U and W are obtained by computing the eigenvectors of XX^{T} and $X^{\mathsf{T}}X$, respectively. Then, it can be shown that $X = U\Sigma W^{\mathsf{T}}$. After SVD is performed, we obtain the principal components as the vectors of the score matrix T, where $T = XW$.

Again, only difference between Statsmodels' and Sklearn's approach in obtaining the principal components is that in the SKlearn approach, X will be restricted to a subset of the entire sample of observations called the train set. Thus, the principal components obtained from the SVD will be slightly different from those of the Statsmodels approach, but in both cases, they will contain most of the variance of the features.

Granger Causality: Vector Autoregressive Model

The previous regressions test the explanatory power of the features where the time indexing is contemporary. The rows of the regression data set we used in the previous regressions was denoted $\{y_t, x_{i,t}\}_{i=1,...,6,t=1,...,299}$, which means that we tested how the features *correlate* with the portfolios month by month rather than test whether they *caused* the variation. Recall that the main argument for using investor sentiment measures as features is the hypothesis that they might affect next month's portfolio returns, so it makes sense to test this using a Granger causality test. The model used to test this is a vector autoregressive model of order p, denoted VAR(p). The order refers to how many lag variables are used, and it is selected by the Akaike Information Criterion (AIC). AIC is a model selection estimator that maximizes the model fit evaluated at the maximum likelihood estimates (MLE) of the parameters, but penalizes for the number of lags to be used in the model (Akaike, 1974). Rather than selecting the number of lags to use in the model arbitrarily, it select the number of lags while still capturing most of the variation in our data set. If we let $\hat{L}$ be the maximum value of the likelihood function for the model, then the information criterion is defined by

$$\text{AIC} = 2k - 2\ln(\hat{L}).$$

Thus, the rows of the regression data set for a VAR(1) model would be $\{y_t, x_{i,t-1}\}_{i=1,...,k,t=2,...,T}$. Hence, our VAR($p$) model is represented as

$$y_t = c + A_1 y_{t-1} + A_2 x_{t-1} + \cdots + A_p x_{t-p} + e_t, \tag{3}$$

where each y_i is a vector with k observations and each A_i is a $k \times k$ matrix. The full matrix notation is then

$$\begin{bmatrix} y_{1,t} \\ x_{2,t} \\ \vdots \\ x_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,k} \\ a^1_{2,1} & a^1_{2,2} & \cdots & a^1_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{k,1} & a^1_{k,2} & \cdots & a^1_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ x_{2,t-1} \\ \vdots \\ x_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,k} \\ a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{k,1} & a^p_{k,2} & \cdots & a^p_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ x_{2,t-p} \\ \vdots \\ x_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix}.$$

Executing this routine remains somewhat similar; We define our regression data set, and then select the VAR model provided in the Statsmodels library. We set it to use AIC to compute the optimal lag length, and let it loop through all features successively. For each feature, it will find the optimal order and run a new VAR(p) accordingly. However, recall from the definition of Granger causality that the universe of information should contain as much relevant information as possible. Then we test whether adding the sentiment features to that universe will improve the predictions. For that reason, we include the market risk premium (MRP) in our regression data set, along with the portfolio returns and sentiment features. Given CAPM it should be fair to assume that MRP contains a significant amount of information. Lastly, the VAR model in Statsmodels is equipped with a Granger causality test submodule, which makes it easy for us to perform them. We simply run the regressions and use their outputs as inputs for the causality test submodule, along with our instructions to test for the isolated effects of adding the sentiment features. Table **??** shows the p-values of these tests.

Machine Learning Approach

In python, Statsmodels is generally the most frequently used module for traditional statistical analysis, as it provides many outputs for in-depth analysis, such as summaries, T-stats, log-likelihood and so on. The GLS and VAR regressions earlier were all performed using Statsmodels classes, as the main goal was just to illustrate the relationship between the features and portfolios. However, the downside of the traditional approach is that results may be subject to *sampling bias*. In other words, the results we obtained were all in retrospect and we cannot be confident that the features would perform similarly had we analyzed them, say, in real-time. Alternatively, one can use the scikit-learn (SKLearn) package, which contains a wide array of both supervised and unsupervised learning modules. The major advantage of using machine learning modules is that we can fit the regressions on a subset of the data and test them on a different, 'unseen', subset. This partitioning approach is known as train-test splitting. Moreover, we can validate our models by cross-validation, which will re-split the data in new partitions and conduct the same regressions again. This way, we can be more confident in how generalized our models really are (Bishop, 2006).

Here, the matrix X of features will only contain past values. In other words, it will be our representation of the universe of information U. Similar to the VAR model, we include as much relevant information as possible and then see if adding the selected features to that universe will improve the predictions. However, there is no Thus, a neat way to isolate the effect of the features will be to first regress past values of the market risk premium (MRP) and respective responses, on the current values of the respective responses. Then, we repeat these regressions, but this time looping through and adding one feature along with the lagged MRP and responses. This way, we can easily compare the prediction error variances with and without the features. Let U be the universe of relevant and available information and let X' and X be subsets of that universe such that $X' \subset X \subseteq U$. Now, let $X'_i \subset X'$ be restricted so that it will only contain lags of MRP as well as lags of the respective responses i. For any $i \in \{1, \cdots, 6\}$, and including all p lagged series chosen by AIC, we have that $\{response_{i,p}, MRP_p\} = X'_i \subset X'$. Now, let $X_{i,j} \subset X$ be a superset of X'_i that also contains the lagged time series of the investor sentiment features. For simplicity, let's call them $feature_j$. Including all p lags again, then for any combination of $i, j \in \{1, \cdots, 6\}$, we have that $\{response_{ip}, MRP_p, feature_{jp}\} = X_{i,j} \subset X$. Lastly, let Y be the set containing the temporal vectors of our responses.

Although they might not be the most popular modules in machine learning, SKLearn ensemble algorithms contain a collection of linear regression models as well. CITATION NEEDED. The selected models in this paper are OLS, Ridge, Lasso and Random Forest. Unfortunately, there are no GLS or VAR model algorithms in the SKLearn library (Pedregosa et al., 2011), and constructing them from scratch is beyond my python skills. For that reason, the existing OLS, Ridge and Lasso regression algorithms are chosen to compare with the GLS regressions - recall that the main reasons for the ensemble approach in this paper is; 1) To see how the features perform out-of-sample, and 2) To select the features that contribute the most to the model.

OLS, Ridge, Lasso and Random Forest

SKLearn's LinearRegression algorithm corresponds to the traditional OLS regression. It takes no hyperparameters and simply fits a standard linear regression with coefficients $w = (w_1, ..., w_p)$ to minimize the residual sum of squares between the observed targets in the data set, and the targets predicted by the linear approximation. The mathematical representation of the optimizing problem is:

$$\min_w ||Xw - y||^2.$$

This method is identical to the methodology mentioned earlier, except that here the vector of coefficients is denoted by w instead of β to make a distinction between the classical and the machine learning approach. Note that we are not taking into account the heteroscedasticity issue we addressed earlier, but we make sure to save the residuals of the predictions in case there seems to exist a non-constant variance.

Recall that the recurring and important issue that applies to most of machine learning algorithms is that fitting the regression to the training set and then predicting on unseen data runs the risk of *overfitting*. The Ridge and Lasso regressions address this overfitting problem of OLS by imposing a penalty on the size of the coefficients. This is otherwise known as regularization. Their coefficients minimize a *penalized* residual sum of squares. Ridge regression uses L2 regularization and hence takes on the optimization problem:

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_2^2,$$

while Lasso uses L1 regularization. It looks quite similar, but has a few important distinctions:

$$\min_w \frac{1}{2n_{\text{obs}}} ||Xw - y||_2^2 + \alpha ||w||_1.$$

The subscripts denote what type of norm the respective regularization methods use. L2 regularization solves the optimization problem with respect to a *Euclidean distance*, while L1 regularization minimizes with respect to a *Manhattan distance*. The difference can be summarized:

$$\begin{aligned} ||w||_1 = & \quad ||w_1|| + ||w_2|| + \cdots + ||w_N||, \\ ||w||_2 = & \quad \sqrt{||w_1||^2 + ||w_2||^2 + \cdots + ||w_N||^2}. \end{aligned}$$

Another important distinction between the two regularization methods is that UNFINISHED SENTENCE

Training, Cross-validation and Testing

Now that we have an understanding of the models, how do we use them? How should we split the data and implement the models?

Most of the machine learning literature deals with classification rather than regression. CITATION. This allows for splitting methods where the data is randomly shuffled before train/test indices are split. However, in time series regressions, it wouldn't make a lot of sense to train the model on sets without respect to sequentiality. For example, suppose we have a small training set containing observations from February 1994, December 2001, and March 2016, while our test set only contains observations from March 2005. By fitting the model to the training set, we would train it using observations 11 years ahead of the test observations. This is not feasible in real life, so we need to use a different strategy. The TimeSeriesSplit function provides train/test indices to split time series data samples that are observed at fixed time intervals. In each split, test indices must be higher (i.e. more recent) than before. Another approach, from now on called the 'manual' approach, is to simply restrict T_{train} observations as the training set and then test on observations in T_{test} where the ladder's indices are higher than the former's indices. After experimenting with different splitting strategies, the best results were obtained using TimeSeriesSplit for cross-validation and hyperparameter tuning and then re-validate successively through the manual approach. This is somewhat confusing, so let's explain this a little further: Suppose we restrict the first 200 observations of all variables as training data, i.e. all data up to and including September 2010. We want to train the algorithm (say Ridge) on this data to predict the next portfolio returns, i.e. observation 201 or October 2010. We then continue successively through all observations, using the first 201 observations to predict observation 202 etc. Naturally, the first problem we face is specifying the best parameters to the algorithm; Should the strength of regularization remain constant throughout all of these splits or could we tune it so that we always use the hyperparameter that optimizes the training score?

WHY I CHOOSE EXPANDING WINDOW SPLITS

INTRODUCE LinearRegression, LogisticRegression, RIDGE, LASSO, AND FOREST STRATEGY: FEED ALL FEATURES AND FILTER OUT THE UNNECESSARY ONES

Results

Due to the number of regressions made, interpreting the results one by one will not be done here. Instead, a summary of the results is shown in Table 3. The coefficients of each regression tells us something about both the directional relationship between each feature and the portfolios, as well as their significance. The asterisks denote what level the estimated coefficients are significant. A positive coefficient indicates that the feature varies *with* the portfolio returns, while a negative one indicates that the feature varies *against* the portfolio returns. For example, the negative coefficients for $cefd$ indicates that an *increase* in the closed-end fund discounts, the portfolio returns would on average *decrease*. Note also that the coefficients for $cefd$ are slightly more significant for small stocks than for big stocks. This is consistent with Lee et al. (1991). Another take-away from this figure is the obvious power of *vixret* - out of the selected features, the change in the fear guage seems to perhaps be the best at explaining the variation of relatively risky equity-only portfolios.

Table 3: Univariate GLS Results

	cefd	NONPNL	CPNL	NCPNL	OI	vixret
SMALLLoBM	-0.0175***	0.0148	-0.0393*	0.0119*	-0.0367*	-0.1897***
ME1BM2	-0.0139***	0.0124	-0.0347**	0.0124**	-0.0260	-0.1560***
SMALLHiBM	-0.0148***	0.0112	-0.0337**	0.0129**	-0.0246	-0.1466***
BIGLoBM	-0.0084**	0.0106	-0.0234*	0.0074	-0.0001	-0.1414***
ME2BM2	-0.0095**	0.0137*	-0.0310**	0.0106**	-0.0078	-0.1358***
BIGHiBM	-0.0111**	0.0162*	-0.0393***	0.0137***	-0.0128	-0.1381***
* $\Rightarrow \alpha = 10\%$	** $\Rightarrow \alpha = 5\%$	*** $\Rightarrow \alpha = 1\%$				

References

Akaike, H. (1974). A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Das, S. R. and Chen, M. Y. (2007). Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9):1375–1388.

De Bondt, W. F. and Thaler, R. (1985). Does the stock market overreact? *The Journal of finance*, 40(3):793–805.

De Long, J. B., Shleifer, A., Summers, L. H., and Waldmann, R. J. (1990). Noise trader risk in financial markets. *Journal of political Economy*, 98(4):703–738.

Dwyer Jr, G. P. and Wallace, M. S. (1992). Cointegration and market efficiency. *Journal of International Money and Finance*, 11(4):318–327.

Engle, R. F. and Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438.

Hiemstra, C. and Jones, J. D. (1994). Testing for linear and nonlinear granger causality in the stock price-volume relation. *The Journal of Finance*, 49(5):1639–1664.

Jain, P. C. and Joh, G.-H. (1988). The dependence between hourly prices and trading volume. *Journal of Financial and Quantitative Analysis*, 23(3):269–283.

Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *The Journal of finance*, 45(3):881–898.

Kara, Y., Boyacioglu, M. A., and Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert systems with Applications*, 38(5):5311–5319.

Kumar, M. and Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*.

Lee, C. M., Shleifer, A., and Thaler, R. H. (1991). Investor sentiment and the closed-end fund puzzle. *The journal of finance*, 46(1):75–109.

Lo, A. W. and MacKinlay, A. C. (1990). When are contrarian profits due to stock market overreaction? *The review of financial studies*, 3(2):175–205.

Malkiel, B. G. (1999). *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company.

Malkiel, B. G. and Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417.

Niarchos, N. A. and Alexakis, C. A. (1998). Stock market prices,'causality'and efficiency: evidence from the athens stock exchange. *Applied Financial Economics*, 8(2):167–174.

Patel, J., Shah, S., Thakkar, P., and Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Samuelson, P. A. (1965). Proof that properly anticipated prices fluctuate randomly. *Management Review*, 6(2).

Smirlock, M. and Starks, L. (1988). An empirical analysis of the stock price-volume relationship. *Journal of Banking & Finance*, 12(1):31–41.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3):1139–1168.

Tetlock, P. C., Saar-Tsechansky, M., and Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467.

Tsai, C.-F., Lin, Y.-C., Yen, D. C., and Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing*, 11(2):2452–2459.