Prediction of Stock Price Movements Using Options Data

Charmaine Chia
cchia@stanford.edu

Abstract— **This study investigates the relationship between time series data of a daily stock returns and features describing the options market based upon the underlying stock. Linear regression was found to be a poor model for predicting a given day's return from returns and options features of the past two days. An alternative approach was attempted by smoothing the returns data using a 26-day exponential moving average (EMA), pre-processing selected options features, and approaching the data as a classification problem. Decision stumps boosting with 10-fold CV was applied to different sets of features to predict EMA returns, with 11.5 % being the lowest average training error achieved. While the EMA return of the previous day was by far the most predictive feature, useful signal was also found in the options-related features. Finally, a new approach to the original regression problem was attempted using the boosting margin as the independent variable. This gave a MSE comparable to the best linear regression performance, and a classification error rate slightly better than that achieved through applying decision stumps boosting to the raw returns data.**

I. INTRODUCTION

Signal detection in finance remains a difficult topic in machine learning, especially for practical applications like price prediction. Successive points in a time series are not necessarily independent and identically distributed, so predictions of a dependent variable's future value need to take into account past values as well as independent variables. Furthermore, financial asset returns often non-normal and display non-ergodic patterns, which can lead to overfitting when standard assumptions are applied. The signals which are easier to detect often are useless as markets drive most to equilibrium price, such that trading on them becomes unprofitable. Often, signal is drawn from data about the underlying assets. For mortgages, we look at characteristics of borrowers, for companies, debt to equity ratio. Hundreds of analysts are paid to develop theories about individual companies and trade on them.

With this in mind, I attempt to use options data to predict stock returns. An option is a contract sold by one party to another, offering the buyer the right to buy or sell an underlying asset at an agreed upon price during a certain period of time. The right to sell is known as a "call" and the right to buy a "put". The agreed upon price is the "strike" (K), to be distinguished from the price of the options contract itself (V). Options can be thought of as bets on the underlying stock price at a given point in the future. The intuition behind this study is that certain aspects of options market behavior could reflect movements of "informed investors". Relevant options features relate to the puts and calls traded and implied volatility. Implied volatility is the *perceived* future volatility of the underlying—a key input into options pricing models, most famously, Black Scholes, determining what contracts are worth.

II. DATA

The data analyzed for this report consists of the stock prices and options data of 57 healthcare companies, over the period from 1/3/2007 to 12/4/2014. From the stocks data, time series of returns was calculated for each company using the formula:

$$Return = \frac{Close\ price\ today\ - Close\ price\ yesterday}{Close\ price\ yesterday}$$

The returns time series can be smoothed to capture the broader trends in stock behavior. This can be done by taking the simple moving average (SMA) over a given interval of days, the exponential moving average (EMA) where the later days in interval are given more weight, and a Gaussian moving average (GMA) where the days in the middle of the interval have highest weight.

The raw options data comprises 39 features relating to the volume of call and put contracts traded each day and various parameters derived from Black Scholes. The data was further split into total, at-the-money, in-the-money or on-the-money contracts, and the relative price of puts to calls, characterized by the put-call parity deviation (PCPdev). It also includes variables associated with the implied volatility, its spread and skew, adding up to a total of 39 features. Given the large number of potential predictors, many of them highly correlated with each other, the question we seek to answer is, which features are most important for predicting the future price of the underlying stock (or equivalently, future returns), and what is the best machine learning model for doing this.

III. LINEAR REGRESSION

As a prelude to building the learning model, the data was studied through visual plots to get a sense for any obvious correlations between the returns data (the outcome variable Y) and individual options features (X_k, $k \in [1:39]$). Both the returns and options data were

smoothed by applying a 30-day simple moving average. Scatter plots of both raw and smoothed data were generated, treating each day as a separate data point ($Y^{(t)}$, $X_k^{(t)}$). This confirmed that the data is was not really normal, with larger tails. However no strong correlation was immediately observable with any of the features. Figure 1 shows the example of 'Returns' vs 'Put volume', for company AET.

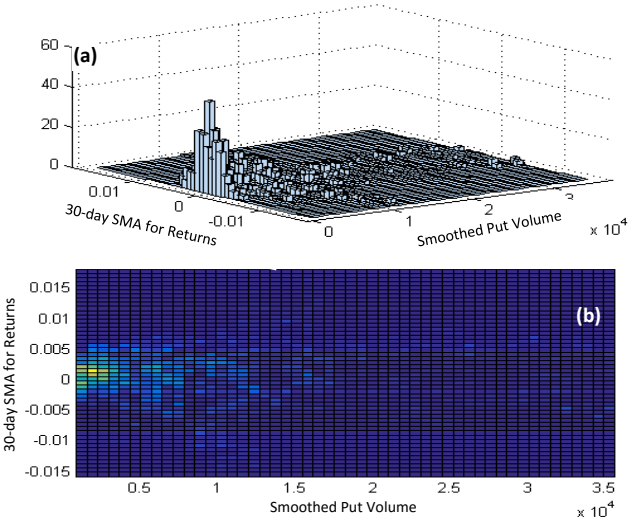

Fig. 1: (a) Histogram & (b) Intensity map of SMA returns vs Put volume

To ascertain if highest cross-correlation between the outcome and independent variable time series occurred at a lag other than $t_{lag} = 0$, cross-correlation plots were generated for raw and GMA-smoothed returns and returns rolling variance versus all 39 options feature time series. The idea is that if temporal trends in the options data do indeed forecast trends in returns, the greatest effect might only occur after a few days, and it would be important to capture this in the regression model. An example is shown in Figure 2.

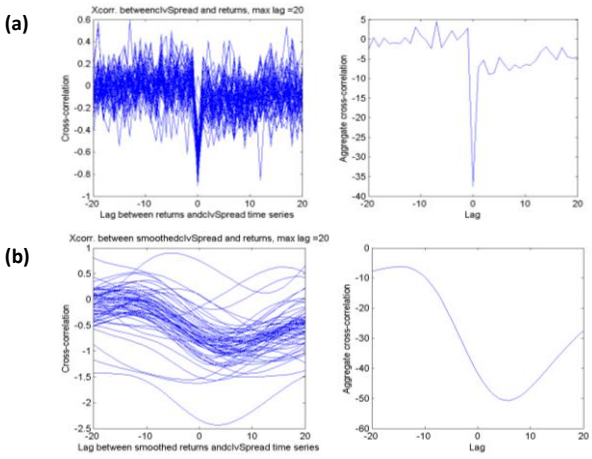

Fig 2: Cross-correlation of returns and implied volatility spread, for (a) raw & (b) smoothed time series

In general the highest cross-correlations were seen at lags of 0 to 5 days. Data smoothing was found to significantly damp out correlation peaks and shift the position of turning points. Unfortunately, the exact interpretation of the plots was not clear; nonetheless based on the average result, we build a rudimentary linear model that regresses $Y^{(t)}$ against $Y^{(t-1)}$, $Y^{(t-2)}$, $X^{(t-1)}$, $X^{(t-2)}$—a total of 80 independent variables. This attempts to capture the dependence on a given day's return on past days' performance and options features. A good result was not expected as far more factors determine the returns of a stock, but this would potentially be illuminating as to what variables can be dropped. Elastic net with $\alpha = 0.5$ was used, with regularization helping to root out potentially misleading predictors. Note however the difficulty of comparing the relative importance of different features based solely on the magnitude of coefficients found, as the independent variables are not standardized (though that comes with problems of its own). Figure 3 shows the results of the regression on one company; Figure 4 is a scatter plot of the results for all 57 companies, summarizing the maximum coefficients (at λ_{min}) for all 80 features and an intercept term, for the outcome variable of (a) raw and (b) smoothed returns. The green stripes highlight the variables which do not have consistently zero (or close to zero) coefficients. These include the previous days' returns, implied volatility related features, and PCPdev-related features. However, a wide range of coefficient values were found, sometimes of opposite polarity. If we assume that a given feature should have a roughly consistent effect on the returns, we would expect its coefficients to have similar values even across different companies. As such, it would seem like linear regression on the unprocessed features is not particularly useful or accurate for modeling returns.

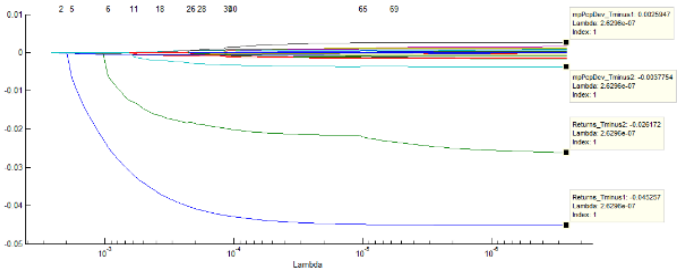

Fig. 3: Elastic net results for $Y^{(t)} \sim Y^{(t-1)}, Y^{(t-2)}, X^{(t-1)}, X^{(t-2)}$

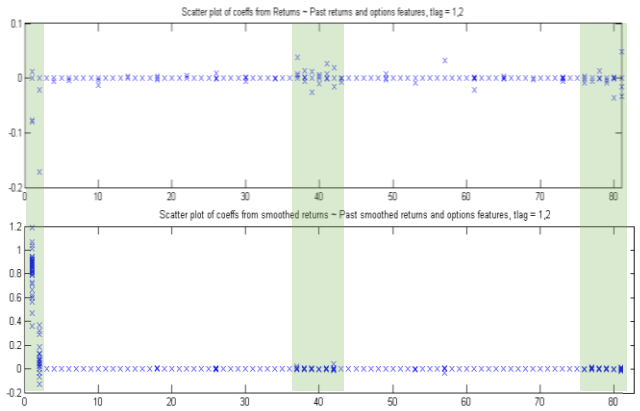

Fig. 4: Scatter plot of regression coefficients of 80 independent variables (+ intercept) for 57 companies

IV. CLASSIFICATION

Given the limited success of linear regression, it is worth checking if the problem can be simplified to a classification one, where we attempt to find signal in the features to predict if returns on a given day will be positive or negative. Secondly, we ask if more predictive features can be found by processing the features from the raw data. Finally, we look beyond linear hypotheses for predicting outcomes.

One indicator that traders have used to gauge market direction is the put-to-call ratio, or PCR. This is obtained by dividing the volume of puts traded by calls traded on a given day. Typically, traders buy stock options to hedge their underlying equity positions, lending credence to the notion that PCR might indicate market sentiment, which in turn might predict market performance. Figure 5 shows historical data from Jan 1997 to May 2002 for Chicago Board Options Exchange PCR (equity-only) values against the S&P 500 closing prices. The dotted lines indicate that an increase in PCR values was followed by declines in the S&P 500, and vice-versa.

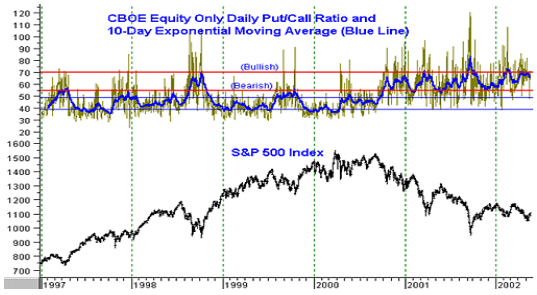

Fig. 5: CBOE PCR and S&P 500 time series

Whether or not the PCR for a specific stock predicts its performance is a slightly different case. To better visualize changing trends in PCR and returns, a 26-day exponential moving average (EMA) was applied to both time series. Further, since market movement is signaled by *changes* in PCR, the daily fractional change in PCR was calculated using a similar formula as that for returns.

3D scatter plots of returns two other options features on a given day were made, and the points color coded according to whether the returns on the next day were positive or negative. Figure 6 shows scatter plots of the returns, PCR fractional change and PCPdev, separated according to the labels of (a) raw returns; (b) 26-day EMA returns. It is clear that the space occupied the points with each label almost entirely overlap in the case of raw returns due to their noisy nature, while there is some separation (though still considerable overlap) when the labels depend on the smoothed returns. Linear, quadratic and RBF kernels were used to separate the labels using SVM. With the number of iterations set to 15000, no convergence was found under the default settings, but by allowing the KKT violation level to be increased to 35% in the linear and quadratic kernels, and 15% for the RBF kernel, classification boundaries as shown in Fig. 6bii were obtained. While this indicates some utility of the method, SVM with these kernels is still not ideal for our data due to high overlap.

An approach based on thresholding both the PCR and the fractional change in PCR was next attempted. This is based on the idea that correlations in movement of the market and PCR happen mainly when the PCR breaks above or below certain levels that indicate whether the market is 'bullish' or 'bearish'. This can be seen in Fig. 7, where the (b) and (c) show different thresholds applied.

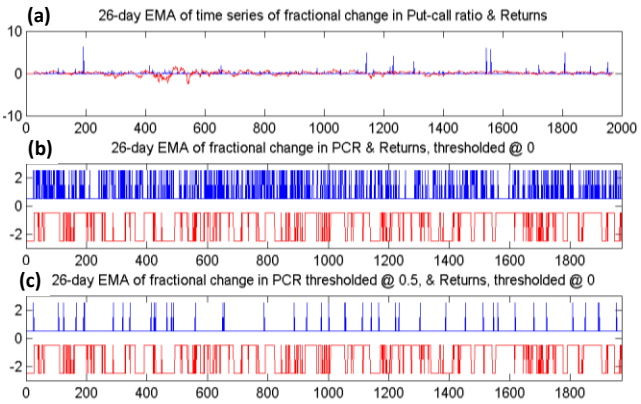

Fig. 7: Time series of Returns (red) and PCR fractional change (blue)

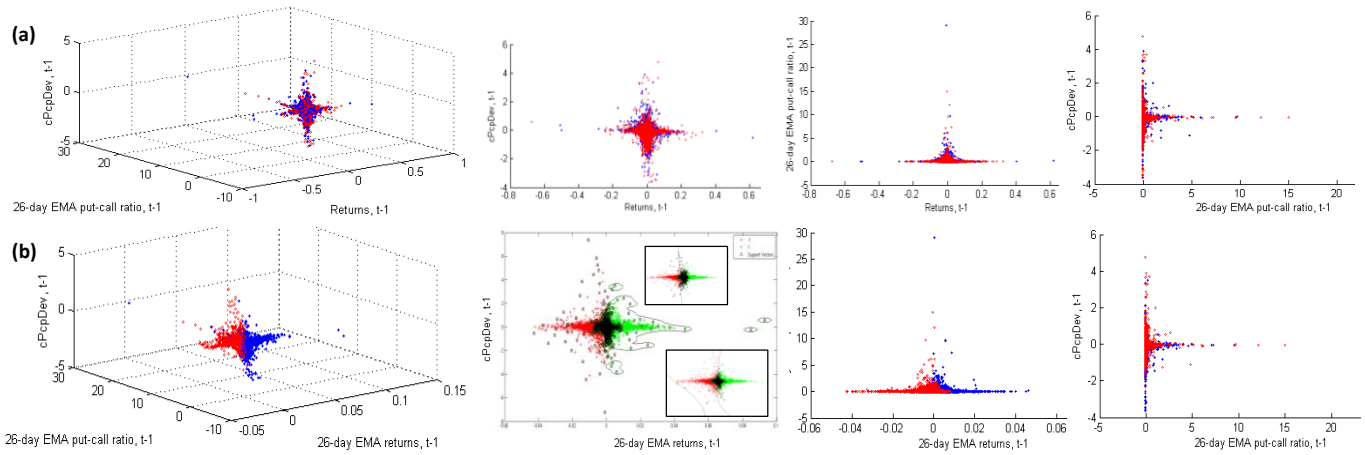

Fig. 6: Scatter plots of (a) Raw and (b) EMA-smoothed Returns at day (t) vs Returns, PCR fractional change & PCPdev at day (t-1)

V. DECISION STUMPS BOOSTING

The use of thresholds suggests the non-linear classification by decision stumps, essentially a one-level decision tree which predicts an outcome based on:

$$\varphi_{j,s}(x) = sign(x_j - s) = \begin{cases} 1 & if\ x_j \geq s \\ -1 & if\ otherwise \end{cases}$$

Individual stumps based on single features are however unlikely to give much better results than chance. The algorithm can be called a weak learner, and we look for some way of combining multiple weak hypotheses to build a much strong classifier.

The ensemble learning method we implement is adaptive boosting (AdaBoost), for which its inventors Schapire and Freund won the Godel Prize in 2003. AdaBoost takes as inputs a weak learner algorithm and a distribution of probabilities $p^{(i)}$ over the training data. It iterates over the hypothesis space of the learner, choosing the hypothesis $\varphi_j(x)$ giving the lowest prediction error on the weighted training data. With each iteration, $p^{(i)}$ are updated to emphasize examples that were wrongly classified. The weights θ_j on the T learners chosen up till current iteration T are updated via coordinate descent to minimize:

$$J(\theta) = \frac{1}{m}\sum_{j=1}^{T} exp\left(-y^{(i)}\theta^T\varphi\left(x^{(i)}\right)\right)$$

After T iterations, the model is based on the weighted sum of predictions of the T learners chosen:

$$\hat{y}^{(i)} = sign\left(\sum_{j=1}^{T}\theta_j\varphi_j\left(x^{(i)}\right)\right).$$

In this case, the weak learner is decision stumps, and the hypothesis space includes all possible features and thresholds for each feature. The following features were included in the feature space, taking into account the observations from the previous sections:

A. EMA returns, days t-1, t-2
B. EMA put-to-call ratio, days t-1, t-2, t-3
C. Put-call parity deviance, days t-1, t-2
D. Implied volatility, days t-1, t-2
E. Implied volatility spread & skew, days t-1, t-2

The outcome variable predicted is the T=26-day EMA return, which can be obtained recursively after initializing the very first interval EMA_0, using:

$$EMA_t = R_t\frac{2}{T+1} + EMA_{t-1}\left(1 - \frac{2}{T+1}\right)$$

Note how we are able to recover a predicted *raw* value of the return for each day, R_t, once we predict EMA_t. 10-fold cross-validation was performed, where data from 5 out of the 52 companies was set aside as the test set each time. The number of boosting iterations T was chosen to be 100. This was repeated for several different combinations of features, with the aim of finding out how predictive different features are. The results and error plots obtained from the experiments are summarized in the Results section.

VI. RESULTS

Table 1 summarizes the average test error rate (over 10-fold CV) after 100 iterations, from the experiments for 8 different choices of feature space. The sets of features included are indexed A to E (described at the end of the last section).

	Returns only	**B, C**	**D, E**	**B, C, D, E**
with A	11.5%	11.4%	11.4%	11.4%
w/o A	48.5% (raw)	37.9%	33.9%	32.8%

Table 1: Results for boosting with different feature sets

Based on the frequency and priority with which certain features were selected by AdaBoost, we can infer how much useful signal for predicting the EMA returns they contain. The top few features appear to be:

1. EMA returns
2. mpIV spread, cIV spread
3. cPCPdev, mpPCPdev

2 and 3 both relate to differences between calls and puts—the difference in implied volatility in the case of 2, and contract price in the case of 3. As such it's not surprising that they contain information about the directionality of the underlying. The prefix 'c' and 'mp' refer to different methods of calculating the each feature.

From the table, we see that the including past EMA returns improves the prediction error dramatically to ~11.5%. This is not surprising given that we would expect a given day's return to depend a lot on the most recent trend, especially after random fluctuations have been smoothed to some degree. As comparison, the same boosting algorithm was also applied to predicting raw returns (from both raw and smooth returns), and the error rate averaged 48.5% —basically not much better than random (see bottom left cell in Table 1).

Interestingly, adding feature sets B, C, D & E did not improve prediction performance once returns were included as a variable. In fact, from the learning curves in Figure 7a-c, we see that not much 'learning' goes on after the first iteration. When EMA returns (feature set A) were not included in the hypothesis space, however, B, C, D, E still give error rates significantly better than chance performance. This indicates that these do contain information that predicts the smoothed returns. Figure 7a – f show the learning curves over 100 iterations for the training and test sets for 10-fold CV, which were summarized in Table 1.

Having shown some success in studying the data as a classification problem, and ascertained which features are most significant, we return to the original regression problem of predicting the magnitude of the return. One approach is to build a hierarchical model, where the first

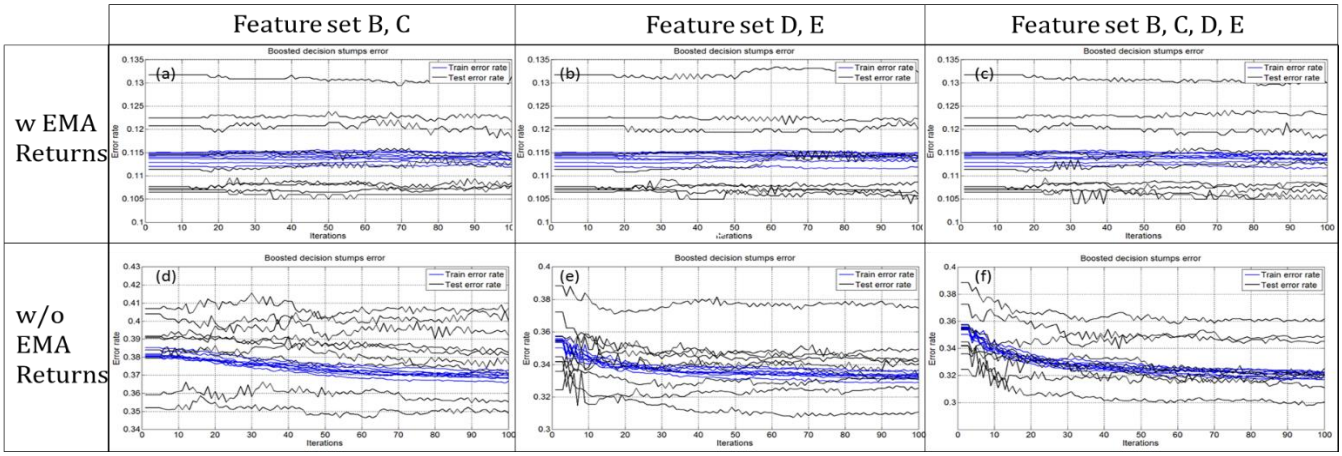

Fig. 7: AdaBoost learning curves for 6 different sets of features over 100 iterations, 10-fold CV. Blue: Training error; Black: Test error

step predicts the direction of the return and the second step predicts its magnitude (given that the first step was reasonably accurate). Regression trees and SVM regression are two methods that could be applied to this.

Here, I briefly suggest another method that builds on the results from AdaBoost. The idea is to use the margin based on which the outcome prediction (± 1) was made as the independent variable in a linear regression to predict the magnitude of the returns. That is:

$$Return(t) \sim \sum_{j=1}^{T} \theta_j \varphi_j\left(x^{(t)}\right)$$

This would only work if a more positive boosting margin, which we would interpret as a higher probability of a $+1$ label, also correlates with larger positive magnitude, and vice versa.

To see how feasible this is, the EMA returns at time t were plotted against the un-normalized boosting margin, as seen in Fig. 8. Evidently, there is a lot of variance about the mean; nevertheless at the extreme ends of the plot, larger absolute boosting margins do seem to predict larger EMA returns. The mean y value for each point along the x axis can approximately be fitted with a 3rd order polynomial as indicated by the red regression line. This can be thought of as the expected value of the EMA return given that the boosting model chosen is accurate. The distribution of points about each value in x can also be further analyzed to obtain the variance given x.

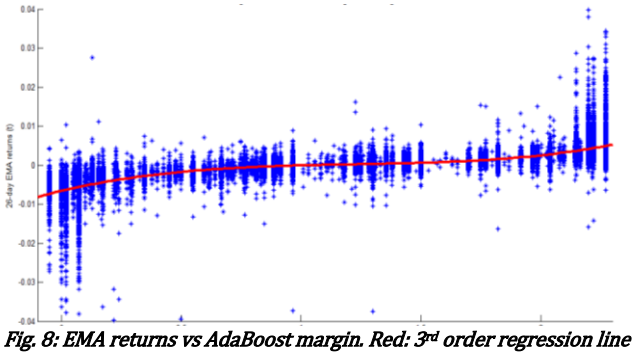

Fig. 8: EMA returns vs AdaBoost margin. Red: 3rd order regression line

As a sanity check, we attempt to convert the predicted EMA return back into a raw return described at the end of Section V. The Mean Squared Error of this prediction over the whole dataset was then calculated by comparing it to the actual return values. This came up to 7.18e-06. As a benchmark, an Elastic Net regression with $\alpha = 0.5$ has MSE ranging from 6.66e-06 to 1.82e-05 as the L1/2 norm constraint is tightened. In other words, this method of regressing on the boosting margin does not seem to be significantly more inaccurate than the most accurate Elastic Net regression. Finally, these raw predicted returns were converted to binary labels and compared with the actual returns labels. The error rate was 44.4%—lower than the 48.5% obtained by prediction using boosting directly on the raw data. While the regression model clearly needs more work and rigorous testing, this is a promising start.

VII. CONCLUSION

Future work could focus on refining the classification model to improve performance, for example by incorporating local weighting into the probability distribution assigned to the data in AdaBoost. Another possibility is to use multi-level decision trees as the base weak learning algorithm, instead of just decision stumps.

Going beyond methodology, the feature space could also be expanded to combine the information present in options data with other variables that are known to be relevant. EMA smoothing could be tried over different intervals of time to find the optimal length. Models could be built attempting to forecast returns further into the future than just one or two days ahead.

VIII. ACKNOWLEDGEMENTS

I am very grateful to Steven Glinert for proposing the original research question, patiently explaining finance concepts that I was new to, helping acquire the data set used, and providing invaluable advice over the course of the project.