Multifaceted Predictive Algorithms in Commodity Markets

Blake Jennings, Hunter Ash, Vineet Ahluwalia

I. Introduction

Commodity price prediction is a notoriously difficult task, and fluctuations in this space have large social, economic, and political consequences, as the recent precipitous drop in oil prices has demonstrated. The ability to predict the movements of commodity prices is highly consequential: a successful prediction algorithm could inform a company on how to hedge against risk or form the basis for a profitable algorithmic trading strategy.

In this project, we employ logistic regression, support vector machine, and Naive Bayes classification algorithms. The input to each model is a collection of time series including technical indicators, macro-market features, and other commodities. Each model outputs a vector $v \in \{0,1\}^N$, in which the n^{th} coordinate indicates whether a given commodity index will have achieved net positive return $n \in \{1,...,N\}$ days in the future.

We begin our exploration with the findings of Jennings' commodity ETF prediction project (Stanford MS&E 447) which demonstrate that models constructed on feature spaces consisting of solely price data and corresponding momentum indicators are insufficient for predicting price movements. This agrees with the conclusions of Kinlay, who ran 1,000,000 different machine learning algorithms using only price data and derivative features and found that none of the models' predictions were distinguishable from noise on out-of-sample data [4].

Kinlay published his 1,000,000 models paper in response to research like that published by Ticlavilca et al [7] and Huang [5] which seems to indicate that machine learning on technical indicators can produce useful and reliable results. Kinlay suspected that the success of many of these researchers is due to sampling bias, and thus performed his exhaustive 1,000,000 model test.

The research by Kase [1] and Jegadeesh [3] demonstrating the profitability of momentum-based trading strategies may have inspired recent attempts by machine learning researchers to use feature spaces consisting only of technical indicators.

Mindful of this prior work, we conduct an iterative process of feature engineering and exploratory analysis to construct an improved input feature space, and show that our model implementations can successfully generalize to out of sample data.

II. Data and Features

A. Feature Sets

In Jennings' project, he pulled data on over 40 commodity exchange-traded funds from Yahoo Finance from 2005 to 2015, and constructed a feature set for each commodity consisting solely of technical indicators on that commodity - primarily simple, volume-weighted and exponential moving averages. Performing recursive feature elimination with logistic regression on these data sets yielded the following average testing and training accuracies:

Training Accuracy	2 day	5 day	10 day	15 day	28 day
Train: 40%	0.609	0.647	0.685	0.715	0.768
Train: 60%	0.592	0.614	0.643	0.663	0.707
Train: 80%	0.573	0.594	0.616	0.634	0.665

Testing Accuracy	5 day	10 day	15 day
Train: 60%	0.527	0.493	0.516
Train 70%	0.495	0.504	0.501
Train: 80%	0.508	0.492	0.513
Train: 90%	0.518	0.519	0.545

Seeing that the model was unable to generalize, we decided to gather daily data comprised of both technical and macro-market features ranging back to 1990 (for all trading days). This data set, obtained from Bloomberg [6], contains 6827 samples of 60 features, including the KC1 coffee futures index, which we attempt to predict. The macro-market features include indices such as the S&P500, US Treasury Yields, and a number of different commodities. The technical features are various momentum indicators which quantify the relationship between recent price changes in a given window and the long term trend of the price of an instrument. Our findings, presented in the following sections, are obtained using this feature space and the direction of the KC1 index as the response variable.

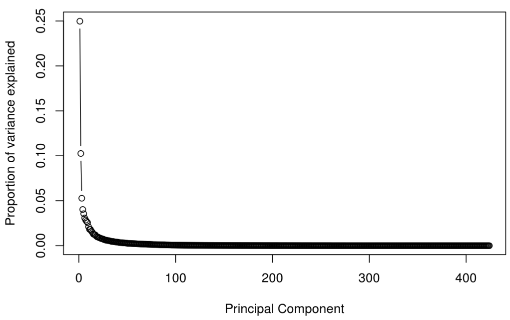
B. Principle Component Analysis

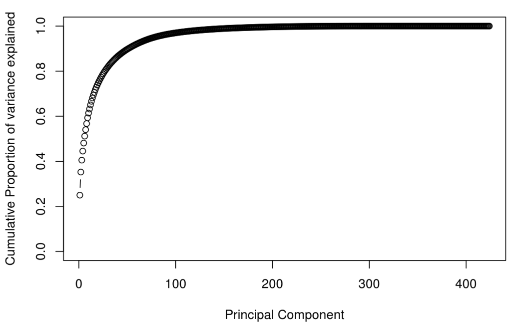
Methodology: To explore our expanded feature space, we perform principle component analysis and look at the top 2 components. For each of these components, we take the 3 most heavily weighted features (highest absolute value of the coefficients).

Results: We found that the first 2 components explained 40% of the total variance. The 3 highest-weighted features for each of them are listed below.

	PC1	PC2
Feature 1	USGG5YR	USSP10
Feature 2	USGG10YR	LEI.BP
Feature 3	FDFD	CONSSENT

The variance explained per component, as well as the cumulative variance explained, are plotted below.





III. Methods

We now present the definitions and specifications of the models we employ. First, we show the how we calculate our response variables.

Computing Response Variables

To compute our response variables, we take the first difference of the KC1 Commodity column:

$$\begin{pmatrix} \vdots \\ p_t \\ p_{t+1} \\ p_{t+2} \\ \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \vdots \\ p_{t+1} - p_t \\ p_{t+2} - p_{t+1} \\ \vdots \end{pmatrix} \equiv \begin{pmatrix} \vdots \\ \Delta_{t+1} \\ \Delta_{t+2} \\ \vdots \end{pmatrix}$$

The net change $\Delta_{t,k}$, k days in the future is then given by

$$\Delta_{t,k} = \sum_{i=1}^{k} \Delta_{t+i}$$

The response variable on day t is an indicator function on $\Delta_{t,k}$ being positive:

$$\text{Direction}_k(t) \equiv \begin{cases} 1 & \Delta_{t,k} > 0 \\ 0 & \Delta_{t,k} \leq 0 \end{cases}$$

A. Logistic Regression

We first investigate logistic regression, in which we fit a hypothesis of the form

$$h_\theta(x) = \frac{1}{1 + e^{\theta^T x}}$$

to our data by minimizing the cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(x^{(i)} - h_\theta(x^{(i)}) \right)^2$$

using the coordinate descent algorithm, which iteratively updates each coordinate of θ according to the rule

$$\theta_k := \theta_k - \frac{\partial J}{\partial \theta_k}.$$

B. Support Vector Machine

We also implement a support vector machine binary classifier, which is the solution of the optimization problem

$$\begin{aligned} \min_{\gamma,w,b} \ & \frac{1}{2}||w||^2 \\ \text{s.t. } & y^{(i)}(w^T x^{(i)} + b) \geq 1 \end{aligned}$$

for $i \in \{1, \ldots, m\}$. This is a convex optimization problem solvable using quadratic programming methods. Since this problem is kernalizable, we try classifying with the following two kernels:

RBF

$$\exp[-||x^{(i)} - x^{(j)}||^2],$$

Sigmoid

$$\tanh\left((x^{(i)})^T x^{(j)} \right)$$

C. Naive Bayes

For our Naive Bayes model, we find the parameters which maximize

$$\mathcal{L}(\phi) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi)$$

or equivalently

$$\ell(\phi) = \sum_{i=1}^{m} \log p(x^{(i)}|y^{(i)}; \phi_x) + \log p(y^{(i)}; \phi_y)$$

For our implementation, we assume a Bernoulli distribution for $p(x|y)$:

$$p(x_j|y) = p(j|y)x_j + (1 - p(j|y))(1 - x_j)$$

where x is a training example and j is the index of one of our features.

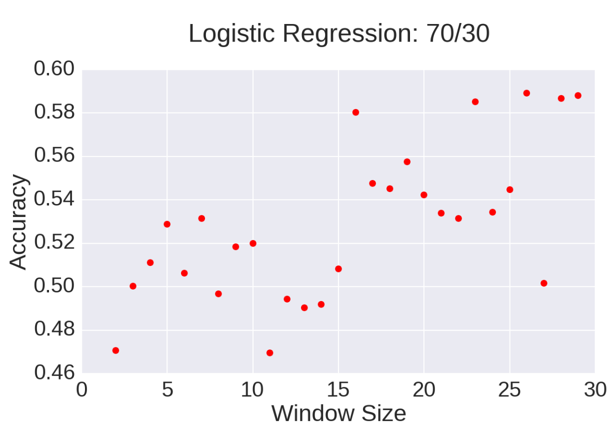
IV. Results

We train 30 models for each of our four classification algorithms, corresponding to every prediction window from 1 to 30 days in the future. We test each of our 120 models using both a 70% training - 30% testing method and a time series version of 8-fold cross-validation.

The prediction windows are the number of days into the future for which we predict a net positive or negative change in the KC1 index. Our time series 8-fold cross-validation method involves first splitting the data into tenths, then training on the first 20% of the data (chronologically ordered) and predicting the next 10%, then training on the first 30% and testing on the next 10%, etc. Therefore, we have 8 training-testing pairs, and the reported accuracy is the average over all pairs. For 70/30, we simply train on the first 70% of the data (again ordered chronologically) and test on the remaining 30%.

A. Logistic Regression

We first test our logistic regression model over all prediction windows, using the 70/30 method:



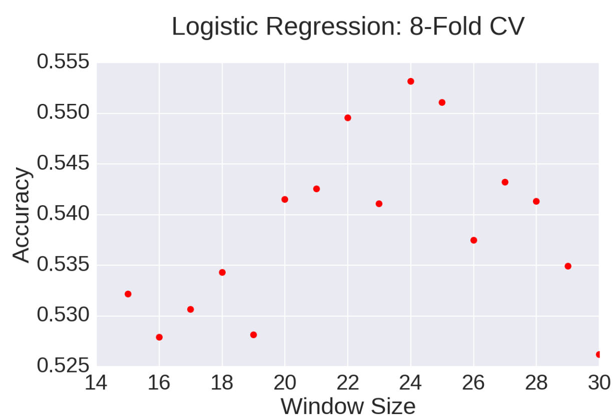
Averaging over all windows, we have the following confusion matrix for this test:

	Predict negative	Predict positive
Actual negative	500	588
Actual positive	329	608

From which we compute

Precision: 50.84%
Recall: 64.89%
Accuracy: 54.72%

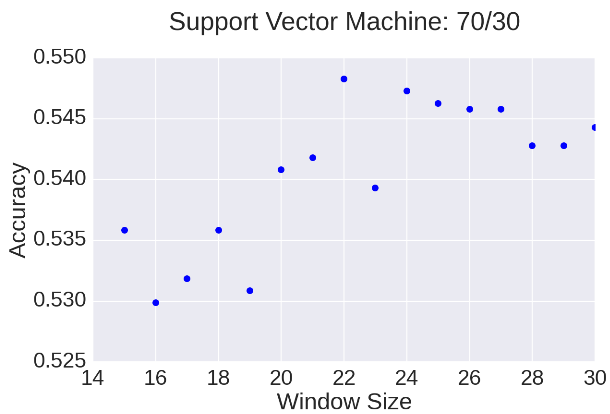
We also obtain an AUC score of 55.41%. We see that our models gain their predictive edge primarily from correctly identifying upward price movements. We also see that models trying to predict windows of less than 15 days do not reliably generalize, and so going forward we restrict ourselves to predicting between 15 and 30 days in the future. 8-fold cross-validation for these windows yields the following accuracies:



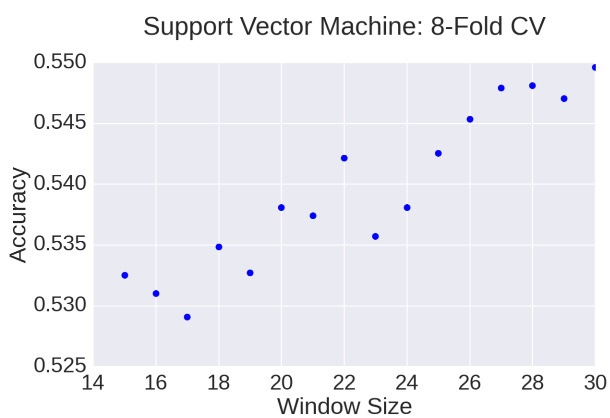
The accuracy generally increases with window size, up to a size of approximately 22-24 days. This general pattern makes sense: increasing the window size smooths out some of the volatility in the price, but eventually starts rendering the information accessible to the algorithm obsolete.

B. Support Vector Machine

We consistently obtain above 53% testing accuracy results (70/30) for our support vector machine with an RBF kernel:



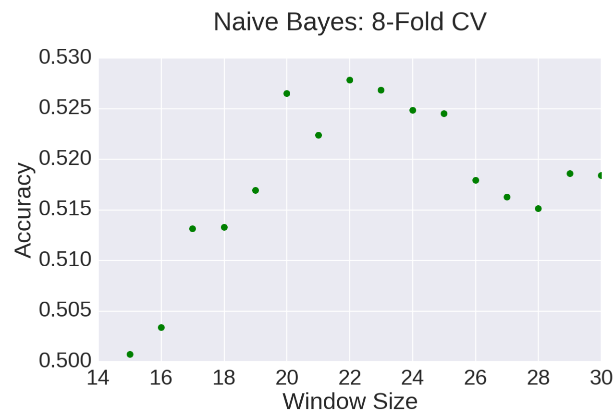
Our accuracies under 8-fold cross-validation are similarly distributed:



Similarly to logistic regression, there is a clear, roughly linear relationship between prediction window size and accuracy. Especially with cross-validation, this trend is more pronounced with the SVM. We also ran our SVM with a sigmoid kernel, obtaining a similar trend, but lower average accuracy. We experimented with a linear kernel, but the increase in run time was so dramatic that we abandoned this route.

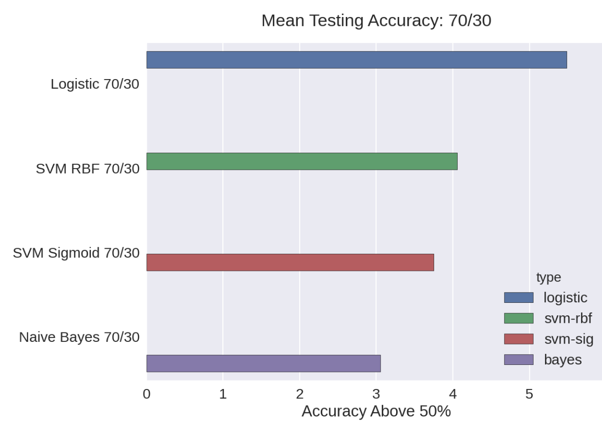
C. *Naive Bayes*

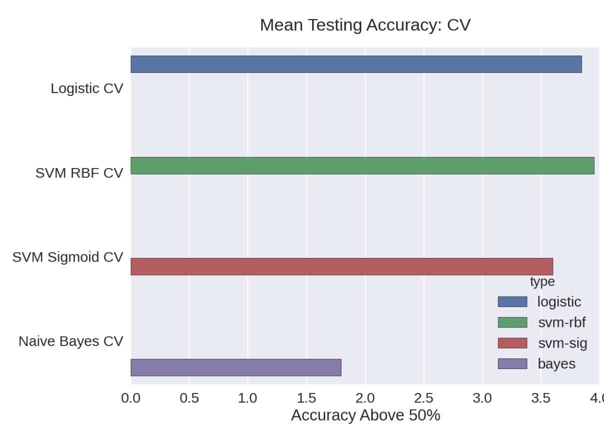
Like our logistic regression and SVM models, our Naive Bayes model consistently attains over 50% accuracy over all windows from 15-30 days. It also has peak accuracy at 22-24 day windows similarly to logistic regression. However, its average accuracy is noticeably lower than our other models. The run time for Naive Bayes was substantially lower than any other model that we experimented with over the course of our analysis. The following graph shows the results of performing 8-fold cross-validation on this model:



D. *Summary*

The following two graphs summarize the mean prediction accuracies for 70/30 and 8-fold cross-validation, respectively:





V. Conclusion

Our analysis confirms that the Bloomberg data set with multiple commodities, technical indicators, and macro-market features has substantially more predictive power for the KC1 coffee futures index than a feature space consisting only of technical indicators. We find that logistic regression gives us the best average testing accuracy, followed by the support vector machine with an RBF kernel. Although Naive Bayes was our least successful model, it still attained consistently above 50% accuracy on windows 15 through 30, with an average accuracy of approximately 53% (70/30 method). In all models, predictions for larger windows tended to be more accurate.

Going forward, we are looking to establish a relationship between the variance and the reliability of different features. Additionally, we hope to employ ensemble methods to improve our accuracy. We are specifically interested in devising a trading strategy to test the profitability of our prediction algorithms.

REFERENCES

[1] Kase, Cynthia A., *How Well Do Traditional Momentum Indicators Work?*, Kase and Company, Inc., CTA 2006.

[2] Becedillas, Gabriel, *Pyalgotrade*, http://gbeced.github.io/pyalgotrade, 2011-2015.

[3] Jegadeesh, N; Titman S, *Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency*, Journal of Finance (48) 1993.

[4] Kinlay, Jonathan, *Can Machine Learning Techniques Be Used To Detect Market Direction? - The 1,000,000 Model Test*, Systematic Strategies LLC,

[5] Huang, D; Jiang, F; Tu, J; Zhou, G, *Mean Reversion, Momentum and Return Predictability*, 2013

[6] Bloomberg Business, http://www.bloomberg.com/

[7] Ticlavilca, A. M., Dillon M. Feuz and Mac McKee. 2010. Forecasting Agricultural Commodity Prices Using Multivariate Bayesian Machine Learning Regression. Proceedings of the NCCC-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management. St. Louis, MO.

[8] Scikit Learn, http://scikit-learn.org/stable/

[9] SciPy, scipy.org

[10] Waskom, Michael, http://stanford.edu/~mwaskom/software/seaborn/