Predicting Market Volatility Using Semantic Vectors and Google Trends

Anusha Balakrishnan
anusha@cs.stanford.edu

Kalpit Dixit
kalpit@stanford.edu

Abstract

Financial trading strategies rely on a knowledge of both the direction and range of market movement (volatility). While existing methods have shown that trends can be predicted using financial indicators or social media, most previous work on predicting market volatility has focused solely on using financial indicators. We hypothesize that social media can more accurately reflect public thought and hence better predict future actions in the market. Specifically, we posit that Google search volume (obtained using Google Trends) for financially relevant keywords often foreshadows movements in the market. In this work, we describe a method to find important financial keywords, as well as a linear regression model that uses Google Trends data to predict market volatility on a per-week basis. Our results show a significant improvement over baselines.

1 Introduction

Existing financial trading strategies often rely on using financial data from the past to make predictions about future market volatility. However, past data simply reflects past actions, but fails to capture any information about public thought - a major predictor of the market's future. Following successful research that uses social data to predict other features of the market, we present an attempt to predict market volatility using Google Trends, which reflects real-time popularity of search terms. Our approach contains two steps: First, we obtain a set of keywords related to the finance domain using semantic vectors. Then, we train a linear regression model to predict the volatility of a future week using as input Google Trends data for those keywords in the previous week.

2 Related Work

Past work on predicting market trends has shed some light on the usefulness of social data. Some recent papers attempt to show correlations between public "mood", as expressed on Twitter, and daily changes in the closing values of the Dow Jones Industrial Average (Bollen et al. 2011) and other financial indicators (Zhang et al. 2011). On the other hand, attempts to predict market volatility have largely used past data to make predictions (Fleming et al. 1998; Corrado et al. 2005), and these methods report varying levels of success. This paper aims to show that social data can have powerful predictive power in modeling volatility, as it does for market movement. More recently, Preis et al. (2013) showed that an invented strategy based on Google Trends yields higher profits than traditional trading strategies like "buy and hold". We add to their work in two significant ways. First, our final keyword set is obtained empirically rather than semi-automatically. Second, we attempt to predict a measure of market volatility rather than directly determine profits using a specific algorithm. This makes our method strategy-agnostic and solidifies the relationship between Google Trends and market movements.

3 Dataset and Features

We are working on the Financial Time Series, defined by the value of the Future of the S&P500 Index traded at the Chicago Mercantile Exchange (CME). The data granularity was one week, where each week is defined as the Open of that week's Monday to the Open on the immediate weeks's Monday. For any Mondays that were holidays, we looked at the first trading day after that Monday.

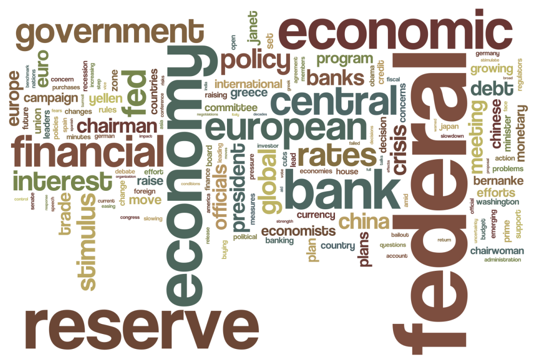

Figure 1: Image representation of financially relevant words found through semantic vector clustering. Larger words appear with higher frequency in the corpus of NYTimes article snippets.

We use all the weeks starting from January 2012-November 2015, spanning 203 weeks of trading. This time period is chosen since the 'Market Regime' during this period is unchanging. Contrast this with including 2008-2009 in our dataset, where those years represent a time of global financial crisis. In the financial world, we have close to zero expectation of one model working on both of these periods. The (Train, Cross-Validation (CV), Test) split was (50%, 25%, 25%) i.e. (100, 53, 50) weeks. Because this is a time series, our CV is defined slightly differently than conventional, in section 4.1.

3.1 Target Value

For each week we measure the high-low percentage $= 100 * \left(\frac{high-low}{open}\right)$.

3.2 Keyword selection

We started with a set of 14 "seed" words (for e.g. "stock market", "federal reserve", etc.) related to finance to query the New York Times (NYT) Article Search API[1], which allows access to the headline, lead paragraph, and abstract of all articles in the NYT from 1851. We searched for articles published after January 1st, 2012 (the start date of our financial dataset) containing one of our seed words and listed under one or more financial news desks.[2] This yielded a corpus of snippets from $\sim$20,000 articles since 2012, which we used to generate semantic vectors using Google Code's word2vec tool[3]. We clustered these vectors into $k = 10$ classes to obtain 10 sets of semantically related words, and filtered out stopwords from these clusters using the standard English stopwords list in Python's NLTK package[4]. We also tried other values of k ($k = 5, 20, 50$), but found that setting $k = 10$ produced clusters that were as semantically general as possible while containing minimal unrelated words.

Since we posit that the search volume for financial terms specifically can predict market volatility, we manually inspected the clusters obtained using word2vec and identified clusters that contained a majority of financial terms. We chose all the words from these clusters ($\sim$850 words) for our keyword set (see Fig. 1).

[1]http://developer.nytimes.com/docs/read/article_search_api_v2
[2] "Your Money", "Business", "Financial", "Business Day"
[3]https://code.google.com/p/word2vec/
[4]http://www.nltk.org/book/ch02.html

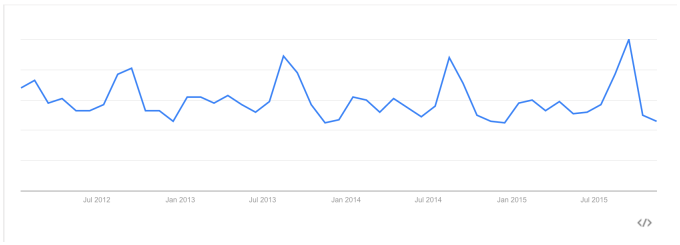

Figure 2: Google Trends from January 2012 - November 2015 for the keyword "labor"

3.3 Features

We obtained the weekly Search Volume Index (SVI) for each word in our keyword set by querying a Google Trends API [5]. The SVI of a keyword represents the search volume of that term *relative* to all other terms. One drawback of using Google Trends is that the SVI for a keyword is based on all searches containing that keyword, and so trend lines often contain spikes caused by searches for unrelated phrases that contain the keyword. Even trends filtered by category ("Finance") can contain noise. As Fig. 2 shows, trends for "labor" within the "Finance" category still show visible spikes in popularity around September every year, coinciding exactly with Labor Day and potentially obscuring trends that are indicative of market movements. We will rely on feature selection methods to remove such keywords. Our final feature set uses Google Trends data from the "Finance" category only.

4 Approach

Our basic prediction problem is: for a given week, predict the hi-low% using Google Trends data from the previous week for all keywords. Formally, for each week w in our dataset, we define $x \in \mathbb{R}^n$ to be the feature set for w containing the SVI values from week $w-1$ for all keywords in the keyword set $K(|K| = n)$. We also define the hi-low% $y \in \mathbb{R}$, and thus our training set consists of m training examples $\{(x^{(i)}, y^{(i)}); i = 1, ..., m\}$.

We train a standard linear regression model with hypothesis function $h_\theta(x) = \theta^T x$, where $\theta \in \mathbb{R}^n$ is the parameter vector. To find the optimal parameters θ, we minimize the standard least-squares cost function $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$ using Newton's Method. To make a prediction on a new week with feature vector x, we simply compute the predicted hi-low% $\theta^T x$.

4.1 Expanding Window Cross-Validation

We wanted our cross-validation to reflect a real-world scenario in which a model is trained using data from all past weeks to make a prediction about a future week. With this in mind, we designed "expanding window" cross-validation: for any given week w, we train our model on all weeks from $t = 1$ to $t = w - 1$, and make a prediction on week w. We then report the cross-validation train error and test error as the average error on the training and the test set (a single week) respectively across all iterations of cross-validation.

4.2 Dimensionality Reduction and Feature Selection

There are two motivations for feature selection: First, our original keyword set is much larger than the size of our dataset, greatly increasing the risk of overfitting. Second, as described in Section 3.3, several of our keywords, although relevant to the finance industry, have trends that are noisy or unrelated to market volatility. For these reasons, we attempt to a) reduce the dimensionality of our feature vectors, using principal component analysis (PCA) and b) remove noisy features by analyzing the correlation between the features and the hi-low%.

[5]https://github.com/dreyco676/pytrends

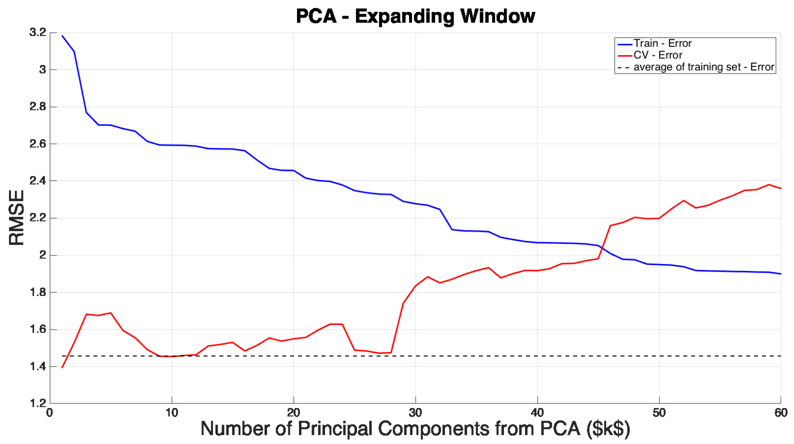

Figure 3: Train vs CV error while searching for the best value of number of Principal Components to use from PCA. As we can see, the CV error never does better than the baseline. Instead, it stays flat with # of components and then diverges.

4.2.1 Principal Component Analysis

We apply PCA to our feature vectors, and use expanding window cross-validation to pick the number of principal components p ($p = 1, ..., n$) that minimizes cross-validation error.

4.2.2 Correlation Heuristic

We hypothesize that only keywords whose trends share a high positive or negative correlation with the hi-low% are useful features. Thus, we computed the correlations[6] between the features (SVI values for all keywords) and the hi-low%. We then ran cross-validation from weeks $w = 101$ to $w = 153$ on increasingly large sets of the top k most correlated features (keywords), $k = 1, ..., n$, and chose our final feature set to be the set that produces the lowest cross-validation error.

5 Experiments and Results

In financial trading, any prediction model which often has low error but occasionally makes large errors proves to be loss-making. To strongly penalize larger errors, we use the Mean Squared Error (MSE) as our metric. While it is typical in time series analysis to underweight old training examples, we consciously did not do that. Since our dataset is small (total of 203 examples), many combinations of features will occur only a few times, and discounting would prevent the model from learning many of them.

As a baseline, we use the mean over the training weeks. Over all constant predictions for the training set, the mean minimizes the MSE i.e. $E[Y_{train}] = \arg\max_{c} E[(Y - c)^2]$.

5.1 Experiments

As explained in section 4.2, dimensionality reduction must be done. We explore the use of PCA and the Correlation Heuristic. Both separately use a single hyper-parameter, called k. For PCA, k represents the number of principal components and for the Correlation Heuristic, it represents the number of features used.

Figure 3 shows the results of using PCA. As we can see, low dimensional PCA projection is worse than the naive baseline. We think the reason for this is that only a few of the 850 features are useful, and are not necessarily orthogonal to the subspace spanned by noisy features. Good features also might not explain most of the variance in the feature space data. Thus, the most important principal components, which are linear combinations of the features, will very likely give a low weightage to the important features. Hence, PCA is ineffective for our problem. We also see that nearly all of the fitting taking place is overfitting.

[6]We used Spearman's Correlation and Rank Correlation, and both agreed.

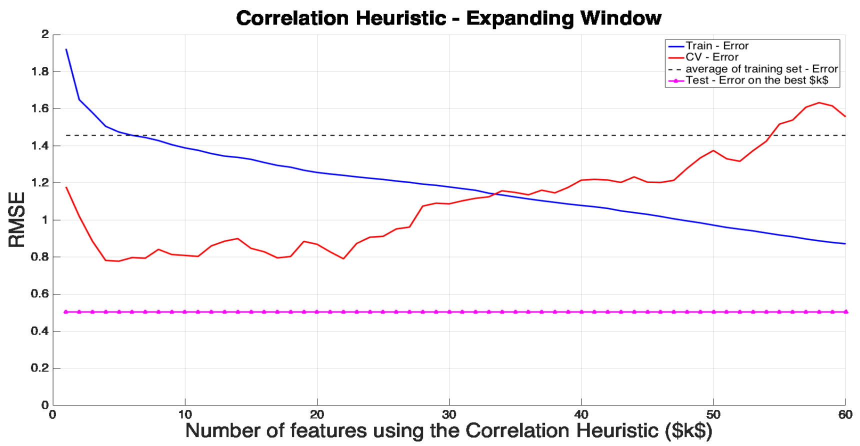

Figure 4: Train vs CV error while searching for the best value of number of features to use. $k = 5$ gave the least CV RMSE of 0.78. The corresponding Test RMSE was 0.51.

Figure 4 shows the learning curve for Linear Regression using the Correlation Heuristic as described in section 4.2.2. It plots RMSE vs # of features used. There are two very interesting observations. Firstly, we can see that CV error reaches a minimum at just $k = 5$, out of 850 features. Qualitatively, this might imply that very few features are actually important. Secondly, for the first half of the graph ($k < 35$), the CV error is lower than the train error. We spent a lot of time in checking our code and look-ahead, and that observation is *not* a mistake. We believe this might be related to the nature of our cross-validation setup. The test set are the last 50 weeks, which is just all of 2015. This implies that the weekly volatility of 2015 is well explained by a model trained on data from 2012-2014.

6 Conclusion and Future Work

We show that market volatility can be reasonably predicted using Google Trends data for financially relevant keywords, and we propose a method for finding such keywords and ranking them by predictive power. We also show that using a very small subset (5 words) of the most correlated keywords produces the best results. A possible extension to this work is the elimination of noise in Google Trends data by removing the effects of unrelated searches. This would further increase the correlation of the trends with actual market movements. Another interesting direction involves pre-clustering the financial data into volatility "classes" (high, medium, low) to determine if we observe different correlations for different classes.

Acknowledgments

We are grateful to ACM Systems for granting us access to proprietary data for this project. We would also like to thank our project mentor, Derek Lim, and Prof. Andrew Ng.

References

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Corrado, C. J., Miller Jr, T. W., et al. (2005). The forecast quality of cboe implied volatility indexes. *Journal of Futures Markets*, 25(4):339–373.

Fleming, J. (1998). The quality of market volatility forecasts implied by s&p 100 index option prices. *Journal of empirical finance*, 5(4):317–345.

Preis, T., Moat, H. S., and Stanley, H. E. (2013). Quantifying trading behavior in financial markets using google trends. *Scientific reports*, 3.

Zhang, X., Fuehres, H., and Gloor, P. A. (2011). Predicting stock market indicators through twitter "i hope it is not as bad as i fear". *Procedia-Social and Behavioral Sciences*, 26:55–62.