Predicting Stock Volatility from Quarterly Earnings Calls and Transcript Summaries using Text Regression

Naveed Ahmad and Aram Zinzalian

Stanford CS224N Final Project Report June 2010

naveed@stanford.edu, aramz@stanford.edu

Abstract

In this paper we explore stock volatility forecasting from quarterly earnings call transcripts of the 30 DOW component stocks. We explore different prediction time frames, with varying sizes of training data, and varying NLP features, such as bags of words, how they affect the predictions, and whether they improve over a baseline using historical volatility alone as the prediction feature. We also summarize the transcripts and explore whether text regression on summaries improves volatility forecasting. We find that with large enough training and test sets, historical volatility together with n-gram features, when used together with stop word filtering, improves over the historical baseline. Summarization was not found to improve model performance. The incorporation of POS adjective tag and handpicked word features slightly improve over the historical baseline.

1. Introduction

We aim to predict future equity return volatility from quarterly earnings call transcripts. Future volatility is of considerable practical interest as it can be indirectly traded through options contracts and is a key component in portfolio optimization. Previous work in text regression by Smith et al.(2009) has shown that simple n-gram features drawn from companies' annual financial reports(10-K filings) improve over a historical volatility baseline in predicting future earnings volatility. We, instead, evaluate the predictive power of quarterly earnings calls in forecasting future volatility.

Publicly listed companies are required by law to file annual 10-K reports with the SEC.
All 10-K filings follow the same format and solely contain information reported by the company. Earnings calls, on the other hand, in addition to giving a broad overview of a company's performance over a quarter, address questions put forward in real-time by independent financial analysts. In this respect, earnings call transcripts better reflect market participants' concerns and likely future expectations than standardized 10-K and 10-Q filings do. This in turn should help us gauge market uncertainty with respect to a given company and forecast future volatility.

Earnings call transcripts can be very long, however, often in excess of 10,000 words. Assuming the average adult reads and comprehends around 300 wpm this amounts to more than half an hour of reading time for each transcript. Processing hundreds of such documents could also be computationally slow. Thus, it would be useful to summarize earnings transcripts. In particular, we use unsupervised content selection based on frequency scores to generate ten-sentence summaries for each transcript which we then train on.

2. Dataset

Our corpus consists of quarterly earnings call transcript of the 30 DOW component stocks from 2006 to 2010. We chose DOW components stock as they are well known big cap companies and represent a diversified portfolio across major sectors. Daily price data was collected from Yahoo Finance in CSV format. We wrote a web crawler to download raw HTML text of all historic transcripts available for a given stock from alphaseek.com. The raw HTML was then further processed to remove HTML tags to produce plain text files, using the HTML Parser API. The final cleaned up corpus consisted of 483 transcripts.

We train on transcripts from 2009 and earlier and test on transcripts from January 2010 and onwards.

2. Volatility

Stock return volatility is a measure of risk. Higher earnings volatility means that a stock fluctuates more in price and hence carries more risk as an investment.

Let $r_t = \frac{p_t - p_{t-i}}{p_{t-i}}$, where r_t is the return, and p_t is the price at time t. Volatility v is computed via sample variance of returns by

$$v^2 = \frac{1}{lookback} \sum_{i=0}^{\text{lookback}} (r_{t-i} - \bar{r})^2$$

We predict volatility instead of price. Previous work in stock price prediction using publicly available data has shown mixed results, arguably because prices already incorporate this information (EMH). Predicting volatility, however, is consistent with economic theory and previous work by Noah A. Smith, Simon Kogan, et al. that shows success in predicting volatility using bags of words from annual company files.

3. Baseline and Evaluation

We report the performance of our predictions using mean square error between the predicted log volatilities and actual log volatilities of equity returns.

$$MSE = \frac{1}{N} \sum_{i=1}^{\text{N}} \left(log\, \text{v}_{pred} - log\, \text{v}_{actual} \right)^2$$

The baseline was to use the historic volatility alone as a prediction feature. Our goal was to see if NLP features such as bags of words improve predictive power i.e. reduce MSE over the baseline.

4. Experiments

We used Support Vector Regression(SVR) to predict volatility from various linguistic features derived from the transcript text. We used the open source java library LIBSVM(Chang, Lin, 2001). Although the RBF and sigmoid kernel function mapping used by SVMs allow us to model volatility as a nonlinear function of the feature space, given that the number of features is much bigger than the number of training examples, it is reasonable to fit a linear model. The test and train data were scaled using a -1 and 1 range using svm_scale utility before being used for training and testing. We use the log of the volatility as the regression training value. The regularization parameter, *C*, used by the SVM for regression, was found for each of the models presented below using 10-fold cross validation.

In addition to extracting features from the unprocessed transcript text files we also generated sentence summaries for each of the transcripts and trained on them. Due to the unavailability of labeled transcript summaries we adopted an unsupervised approach. As a first pass, we scored sentence *j* in document *k* as

$$S(j,k) = \sum_{i=1}^{|s|} (c_{ik}/c_i)$$

Where $c_{i\mathrm{k}}$ is the number of occurrences of word *i* in transcript *k* and c_i is the frequency of word *i* across all transcripts. Intuitively, this score gives greater weight to words that occur often in a document but not as often on average in the corpus. For this reason, names of participating equity research analysts who follow one stock are scored the highest. This gives undue weight to uninformative sentences like "John Doe – VP of Research at Bank A." The format of the transcripts, however, is such that analyst names and their titles precede the core text, making it easy to parse out this information. For each transcript we generate a set of these irrelevant names and titles and assign a score of zero to them in the core text. We also selectively ignore common stop words.[1]

The additive scoring for sentences favors longer sentences, but these were indeed found to be very informative, often directly addressing fiscal and operational risk and uncertainty.

Next, we used term frequency-inverse document frequency (TIDF) to score n-gram x_j in document *d*, in a corpus of *N* documents. We then generated features for these scores from the summarized transcript texts and the transcripts themselves.

- TFIDF: 1/|d| freq (xj:d) * log (N/|{d:freq(xj:d)>0}|. Previous work shows that using TFIDF scores results in the best results. Other criteria such as log1P scores could also have been used to score n-grams.

We performed experiments by varying

- Prediction time frame i.e. look-back and look-forward periods by 20, 40, and 60 days.
- Train set size from 25 to 386 transcripts. The ratio of test data to train data was kept at approximately 1 to 5.
- NLP Models between
  1. Unigram with TF/IDF
  2. Bigram and Unigram with TF/IDF
  3. Unigram and Bigram with Sentence Summaries

[1] http://www.lextek.com/manuals/onix/stopwords1.html

Days	#Test	#Train	Base	Unigram	Bigram	Unigram Summ	Bigram Sum
20	5	25	0.0665	0.0893	0.0886	0.0603	0.081570
40	5	25	0.0772	0.1410	0.1447	0.0693	0.064829
60	5	25	0.1252	0.1638	0.1632	0.1596	0.148450
20	15	75	0.0637	0.1040		0.1335	0.135393
40	15	75	0.0464	0.0881		0.0840	0.071755
60	15	75	0.0645	0.0915		0.0843	0.083942
20	25	125	0.0783	0.0803	0.0817	0.0915	0.093217
40	25	125	0.0684	0.0740	0.0746	0.0874	0.084544
60	25	125	0.0722	**0.0717**	**0.0719**	0.0655	0.063674
20	35	175	0.0679	**0.0667**		0.0940	0.097110
40	31	175	0.0643	0.0743		0.1005	0.094078
60	31	175	0.0653	0.0694		0.0649	0.062576
20	45	225	0.0634	**0.0567**	0.0580	0.0939	0.095536
40	31	225	0.0611	0.0741	0.0747	0.1013	0.098861
60	31	225	0.0577	0.0685	0.0688	0.0733	0.072260
20	51	275	0.0549	0.0589		0.0963	0.099751
40	31	275	0.0636	0.0799		0.0997	0.100878
60	31	275	0.0610	0.0726		0.0743	0.075759
20	51	325	0.0550	**0.0544**	0.0555	0.0951	0.097484
40	31	325	0.0653	0.0760	0.0765	0.0940	0.097174
60	31	325	0.0672	0.0690	0.0690	0.0688	0.072821
20	51	375	0.0558	**0.0500**		0.0885	0.093059
40	31	375	0.0663	**0.0613**		0.0875	0.093604
60	31	375	0.0676	**0.0583**		0.0648	0.067723
20	51	386	0.0568	**0.0501**	**0.0509**	0.0897	0.093613
40	31	386	0.0664	**0.0615**	**0.0627**	0.0867	0.093643
60	31	386	0.0633	**0.0576**	**0.0581**	0.0654	0.068468

Table 1. MSE values from different models. The values reported in bold show improvements over the baseline MSE. Our results show that predictive performance increases with training size and is otherwise very variable when using a small train set.

6. Parts of Speech

We also investigated the use of part-of-speech (POS) tags in forecasting volatility.

Below is an example of a tagged sentence by the Stanford POS Tagger. JJ stands for adjective. An increasing use of adjectives could mean vagueness and uncertainty from the presenter of the transcript. This is consistent with our observation that longer sentences tend to address questions pertaining to risk and hence are more informative. We ran the test with the maximum data in our corpus and found that using the extra feature improved MSE only slightly, but was interesting nonetheless to observe.

“In/IN challenging/JJ and/CC difficult/JJ market/NN conditions/NNS ...”

We incorporated the Stanford POS tagger and added the following 2 features.

- Total Count of Adjectives in a Transcript
- Total Count of Adjectives /Total Words in a Transcript

Days	#Test	#Train	Base	Unigram	+POS
20	51	386	0.056805	**0.050100**	**0.050099**
40	31	386	0.066436	**0.061477**	**0.061467**
60	31	386	0.063283	**0.057638**	**0.057626**

Table 2. Adjective POS features improve the result very slightly using the maximum training data.

7. Hand Picked Words

Reading though the transcripts, a few words occurred more often than others when indicating trouble or unexpected news about the company. We picked words that could be predictors of volatility, to determine whether a feature using those words could improve MSE. We chose the following strings, with the feature being the total count matching these in a transcript: **risk, uncert, challeng, volatil, exceed**.

Table 3. Comparison of Unigram MSE with including hand picked words. A small improvement in MSE.

Days	#Test	#Train	Base	Unigram	+Words
20	51	386	0.056805	**0.050100**	**0.050094**
40	31	386	0.066436	**0.061477**	**0.061475**
60	31	386	0.063283	**0.057638**	**0.057636**

8. Analysis of Results

We observe that with a larger train set the MSE for unigrams is consistently lower than that of the baseline MSE. With training data size of 375 transcripts and upwards, the Bigram Model also improves over the historical baseline. We can only conclude that a much larger training corpus should have been used.

Contrary to our expectations, the MSE of the bigram model was higher than that of the unigram model. We gathered all the unigrams from the transcripts we examined but ran into memory issues when collecting bigrams, ultimately having to reject around 90% of the lower frequency bigrams. It may also be that many of the more common bigrams such as “we are” are actually uninformative and those that do have predictive value contain only one important unigram anyway.

Training on summarized data did not improve MSE and actually yielded worse results than the historical baseline (see Table 1.) It is possible that together with stop words, our summarization filtered out information too aggressively. The generated summaries are, however, qualitatively informative and could be used to assign higher feature weights to in-summary words. Finally, adjectives and handpicked word features marginally improved MSE over the test data.

9. Future Work

Potential future work could include

- Increasing the training size to several hundred stocks.
- Using a different SVM package. With LIBSVM we ran into memory problems when we ran with bigrams using the maximum training data size.
- Devising some heuristic for choosing good bigrams, either through pairings with relevant target words indicative of risk or through dependency on such words in sentences.
- Evaluating the performance of more flexible RBF and sigmoid kernel functions.
- Refining evaluation metrics for summarization accuracy. Given the Q/A format of earnings calls, instead of summarizing the whole transcript sentence by sentence one could summarize each transcript question by question.
- Determining sentence orientation of a transcript by comparing partial mutual information scores between component words and a hand-picked corpus of predictive unigrams [6].
- Using regression trees. The interpretability, flexibility, resistance to outliers, and missing values make regression trees an attractive alternative to SVMs for NLP related tasks. That said, the preliminary trees we built indicated that historical volatilities were much more significant predictors than any of the NLP features we created.

10. References

[1] Noah A, Smith, Shimon Kogan, Dimitry Levins, Bryan R Routledge, Jacob S. Sagi "Predicting Risk from Financial Reports With Regression",

[2] Reza Bosagh Zadeh, Andres Zollmann, "Predicting Market-Volatility from Federal Reserve Board Meeting Minutes".

[3] Mahesh Joshi, Dipanjas Das, Kevin Gimpel, Noah A. Smith, "Movie Reviews and Revenues: An Experiment in Text Regression".

[4] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques".

[5] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[6] Alistair Kennedy and Diana Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters".

[7] HTML Parser. http://htmlparser.sourceforge.net/.

[8] Kristina Toutanova, Dan Klein, Christopher Manning, William Morgan, Anna Rafferty, and Michel Galley. Stanford Log-linear part of speech tagger.
http://nlp.stanford.edu/software/tagger.shtml.

[9] Quarterly Earnings Call Transcripts Obtained from: http://seekingalpha.com/.

[10] Jurafsky, Daniel, and James H. Martin. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition. Prentice-Hall.