Predicting Adverse Disclosures in Corporate Filings

Alex Nocella

19 December 2017

https://github.com/anocella/w266project

Abstract

The purpose of this project is to automatically and algorithmically discover language in corporate filings that might serve as an early warning of poor future stock performance. We compare documents from the same company over time to look for changes in the language, and we correlate those results with stock performance. We include the previous four documents from the company's SEC filings (Form 10-Q and Form 10-K), and we compute cosine similarity scores on vectorized transformations of those documents. In addition, we count words that are sentiment-classified as "positive," "negative," or "litigious" per document as features. We attempt to solve a binomial classification problem of whether the company's stock will out- or under-perform the S&P 500 index between the filing date of the relevant document and the filing date of the next document.

Introduction

Public companies are owned by their equity investors, but they are largely operated without influence from the majority of those investors. The chief officers of these companies are appointed by the board of directors and they, as agents, are responsible for properly disclosing relevant information in publications like the annual report. However, there is little opportunity for investors to validate the contents of these reports, and there are plenty of examples of deceit throughout history that have passed audits (WorldCom, Enron, Waste Management, among others). Detecting misleading statements or half-truths is especially challenging when the only real information available is a written document with ample time to prepare. There are thousands of public companies listed just on United States exchanges, and keeping track of all of their activity requires a great deal of sophistication and a vast number of man-hours (see: the entire investment banking research industry). Ultimately, we address the bottom line for investors: can we predict, using only the language content in a financial statement, whether the stock will perform poorly in the future?

The S&P 500 stock index by itself represents in excess of $20 trillion in equity wealth[1], and nearly $8 trillion of investment company assets are benchmarked against it[2]. Eugene Fama's seminal 1970 paper on efficient markets describes his findings in support of full efficiency of "obviously publicly available information," which we perhaps liberally interpret as a conclusion that there is insufficient evidence supporting the idea of realizing risk-adjusted outperformance using the face-value content of financial statements (Fama, 1970). The efficacy of the efficient market hypothesis is hotly debated still today, and we intend to challenge it with a classification algorithm using what can only be described as obviously publicly available information. Investors are always seeking the ability to generate returns in excess of the S&P 500 index, which is evidenced by the historical standard "2 and 20" fee structure for hedge funds – a 2% fee on all assets under management, and a 20% incentive fee of the profits. The amount of

[1] http://www.businessinsider.com/sp-500-market-cap-crosses-20-trillion-for-the-first-time-2017-2

[2] http://us.spindices.com/indices/equity/sp-500

money and fame at stake for crafting outperforming investment strategies is so large that even the smallest amount of true skill is invaluable. So, naturally, we set out to create such a strategy.

Background

In our review of the literature, we find examples of papers in which the authors could classify documents or financial transactions (credit card transactions, etc.) as fraudulent with accuracies well above naïve models, but they are difficult for us to recreate due to data availability. In the case of identifying company fraud with financial statements, Goel et al. identify 126 instances of companies "accused of fraudulent reporting in the period from 1993 to 2006." Using a bag-of-words approach as a baseline model, they achieve an accuracy of 56.75%. Their more complex model, including features like "percentage of passive-voice sentences," TF-IDF weighted tokens, and frequency of words with positive sentiment improves the accuracy to 89.51% (Sunita Goel, 2010).

Similarly, Humpherys et al. compare different binomial classification models like logistic regression, Naïve Bayes, and SVM on a dataset containing 101 each of non-fraudulent and fraudulent 10-K forms, 69 of which were deemed fraudulent due to "overstatement of revenues" or "combination of overstating revenue and understating expenses." The authors process the 10-K forms with part-of-speech taggers and include features like number of words, number of verbs, lexical diversity, frequency of passive verbs, etc. The best performing model is Naïve Bayes and produces a 67.3% accuracy (Sean L. Humpherys, 2011).

Chen et al. also compare documents that are tagged are fraudulent and non-fraudulent annual reports from companies listed on the Taiwan Stock Exchange. They create a term library to classify terms that are associated with fraud. The authors use part-of-speech tagging and filter stop words, then pass annual statements through a TF-IDF algorithm to sort whether a given term is associated with fraudulent statements. They report an accuracy of 85.25%, which is an improvement of as much as 20% over other text-data classification studies in the literature (Yuh-Jen Chen, 2017).

The literature is dominated by cross-sectional datasets used to create some model of what a fraudulent document looks like in a pool of documents. We find that collecting or creating such a tagged dataset is infeasible for the scope of this paper, and instead use the insights of these works to approach a market-based problem. The premise of our research is that company filings can be benchmarked against themselves across time, which changes the framework from identifying the bad actors in a group to instead identifying linguistic shifts that correlate with stock returns. This conveniently sidesteps the issue of data availability, and additionally allows for tens of thousands of training samples – a stark contrast from the datasets on the order of a few hundred observations in the prior work.

The concept of algorithmically processing financial statements to predict future stock returns is not novel. We draw inspiration from Li (2006). Li finds that an increase of risk sentiment in 10-K filings "is associated with lower future earnings: firms with a larger increase in risk sentiment have more negative earnings changes in the next year." Also, Li proposes a portfolio whose strategy is buying stocks with low increases in risk sentiment while shorting stocks with high increases in risk sentiment. Li concludes that such a hypothetical portfolio could produce more than 10% of annual excess returns. Li's primary feature is a count of the instances of risk- and uncertainty-related words, which the paper describes as a simple, yet effective, technique. Risk sentiment is defined as $\ln(1 + NR_t)$, where NR_t is the number of risk

words in the annual statement, and change in risk sentiment is simply $\ln(1 + NR_t) - \ln(1 + NR_{t-1})$ (Li, 2006).

Methods

Data

The most daunting task of this research was curating the dataset. Li (2006) reports a sample from annual reports (Form 10-K) of 34,180 firm-years of non-financial firms in the Center for Research in Security Prices (CRSP) universe between 1994 and 2005. In contrast, we collect both Form 10-Q and Form 10-K for all current (December 2017) members of the S&P 500 index for as far back as they exist on a per-company basis starting in 1994. We retrieve the forms from the United States Securities and Exchange Commission's Electronic Data Gathering, Analysis, and Retrieval system (EDGAR, https://www.sec.gov/edgar/searchedgar/accessing-edgar-data.htm). The EDGAR system uses a Central Index Key (CIK) identifier issued by the SEC. A list of S&P 500 constituents and their CIK is available at https://en.wikipedia.org/wiki/List_of_S%26P_500_companies. All relevant Form 10-Q and Form 10-K filings are collected, and we discard instances of the same form filed multiple times in one day. After all transformations and initial cleaning, there are 38,954 observations in the dataset, each representing one firm-quarter. We also collect stock prices from the Quandl WIKI Prices database API, which is freely available at https://www.quandl.com/databases/WIKIP. S&P 500 Total Return values are available from Yahoo! Finance at https://finance.yahoo.com/quote/%5EGSPC/history?period1=631170000&period2=1513486800&interval=1d&filter=history&frequency=1d**.** For each filing date, we compute the log return of the stock's price since the last filing date (generally one quarter), and the log return until the following filing date (which is the basis of our target label). The total size of the EDGAR files is well in excess of 200 GB.

Features

With all of the EDGAR data downloaded, we parse the filings in a relatively simple manner. As with the aforementioned papers, we employ the TF-IDF algorithm and use it to vectorize unigram through trigram features in all of the documents on a per-company basis. The forms are preprocessed to remove as much of the machine tagging and metadata as possible. All digits are ignored, and all words are set to lowercase. Stop words are filtered out, and remaining tokens must be at least 3 characters long. The tokens are then filtered through a set of English words (available at http://www-01.sil.org/linguistics/wordlists/english/). Simultaneously, we apply simple count vectorization (again with unigrams, bigrams, and trigrams), as well as count tokens that are tagged as "positive," "negative," or "litigious" (available at https://www3.nd.edu/~mcdonald/Word_Lists.html). For each document, we compute the cosine similarity between its vector and the vectors of the prior four filings, which generally corresponds to the four prior quarters. This similarity measure is calculated for the TF-IDF vectors and the count vectors, creating eight cosine similarity features. We further create new features measuring the change in cosine similarity between each prior filing and the current filing.

We join the document feature dataset with the stock data obtained from Quandl. The stock's excess return, defined as the difference between the return of the stock and the S&P 500 index between the two filing dates, since the last filing date becomes a feature (as well as the direction of the excess return mapped to "true" or "false" for whether the stock outperformed). The target variable is whether the stock's excess return is positive ("true") or negative ("false") until the company's next filing date. We transform the counts of sentiment-tagged words to logs, compute the relative count frequencies as a

fraction of the total number of tokens, and compute time differences for all of these compared to the time-zero filing. There are 60 features altogether after all of these manipulations.

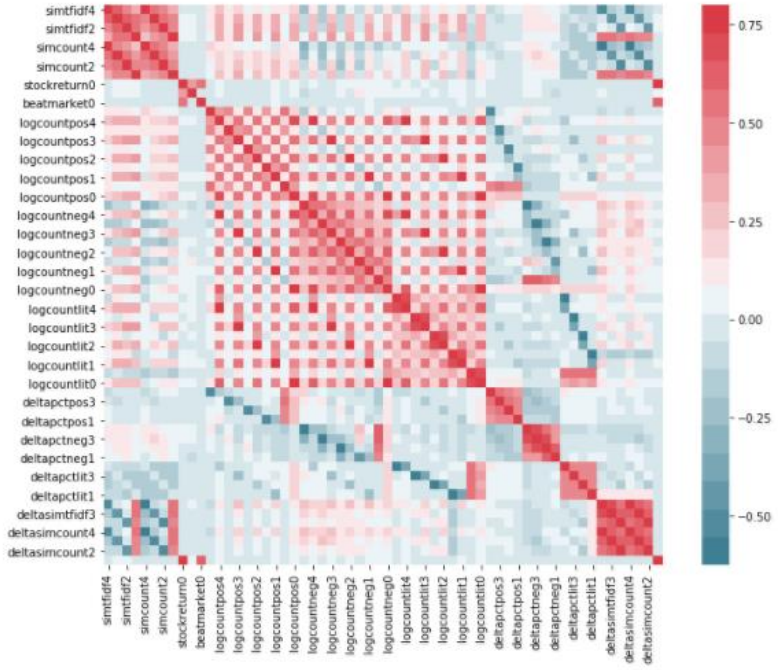

Figure 1: Correlation matrix of all features.

Classification

We treat this problem as a binomial classification (outperform/underperform), in contrast with Li (2006) which uses a linear regression model and targets both the direction of the stock's excess return and its magnitude. This decision is partly out of necessity: we do not control for basically any attribute of the company, including the stock's covariance with the S&P 500 index, where Li (2006) does control for the Fama-French factors and momentum. Without such control variables (the Fama-French model was the most widely accepted equity factor model since Markowitz's Capital Asset Pricing Model, and the Fama-French-Carhart model which adds momentum is the most widely accepted since the original Fama-French paper), the magnitude of excess return would certainly be misestimated. We thus implicitly assume that each stock's average market sensitivity has a factor of 1.0, which must be on average correct because we include all of the constituents of the market index.

We first separate the dataset into training, dev, and test sets. The training and dev sets includes all data through 2014 backward, and the test set includes all data starting in 2015. I made this decision to avoid any possible "look-ahead" bias, which is normally a huge concern in financial backtests – merely knowing what generally happened in markets during validation on held-out data could arguably bias the model. There may be some temporal nonstationarity in the covariates that are not captured as a result, which we do not explicitly examine and which would tend to provide greater confidence in the model output (the test set should ideally provide results as if the observations were never possibly known during model training). The dev set is a randomized 30% subset of the non-test set separated from the training set, and chosen such that the frequencies of outperformance/underperformance are identical (which is not a herculean task given that they start out extremely close to 50/50).

The data are very far from linearly separable. Where the fraud detection papers (Goel, Humpherys) find some success with logistic regression, it is all but useless here (which was a discouraging realization after the arduous data gathering exercise). As alluded to earlier, though, if it were this easy to predict market-beating stocks, then everyone would be rich. Similarly, linear SVM and PCA are out of the question.

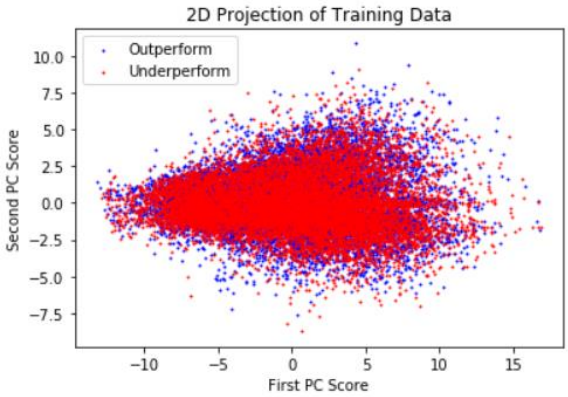

Figure 2: Scatterplot of first two principal components.

Results

We try some nonlinear classification models with nothing to show for it. K-nearest neighbors and Naïve Bayes never perform noticeably better than a baseline model of just choosing the (slightly) more frequent class on the training data, and in many cases KNN performs noticeably worse. Some experiments with AdaBoost and random forest classifiers are mildly promising on the training set, but we cannot get the models to generalize well enough to beat a naïve constant baseline model on the dev set.

In an unanticipated last-ditch effort, we train a multilayer perceptron classifier. Some informal hyperparameter tuning on the solver and activation function pairs leads to the conclusion that Adam and ReLU provide the best results on the dev set (other solvers tested are stochastic gradient descent and limited-memory Broyden-Fletcher-Goldfarb-Shanno; other activation functions tested are logistic and tanh). The MLP classifier's training accuracy is highly sensitive to the number of hidden layers and the number of neurons per layer. We settle on 4 hidden layers, each with 100 neurons, as we find it provides the most robust accuracy and F-score pairs. Performance drops off substantially at both 3 and 5 hidden layers, and 1 hidden layer is expectedly as poor as a linear model.

The model predicts stock outperformance to underperformance at a rate of about 1.66:1 on the dev data (6,112 outperform and 3,690 underperform predictions), and this is a common theme among all of the models we test: the predictions on the entire dev set (and frequently the training set) are not close to evenly split, despite human knowledge that they must be very close. Thus, it appears fruitless to interpret the model probabilities as physical probabilities because they could not possibly describe reality. The distribution of probabilities is such that one-half of the dev set have class probabilities in the range of [0.4, 0.6], only one-fifth of the probability space. We additionally find that the model is nothing close to calibrated: the precision of stock underperformance predictions on the dev set is 51.4% when predicted probability of underperformance is at least 0.6, and the precision rises only to 58.2% when the model probability is at least 0.99 (the full accuracies are 54.4% and 55.4% respectively).

We consider an interpretation of the model results such that predicted probabilities on [0.4, 0.6] are thrown out. Under this conditional case, we find the following results on the dev and test sets (the positive event for precision and recall is stock underperformance):

```
Minimum predicted probability is 0.6
Count prediction == Outperform:          3087
Count prediction == Underperform:        1935
Predicted Out and actually Out:          1737
Predicted Under and actually Under:      994
Conditional Precision:                   0.514
Conditional Recall:                      0.424
Conditional F-measure:                   0.465
Conditional Accuracy:                    0.544
```

Figure 3: MLP classifier results on dev set.

```
Minimum predicted probability is 0.6
Count prediction == Outperform:          1275
Count prediction == Underperform:        163
Predicted Out and actually Out:          652
Predicted Under and actually Under:      97
Conditional Precision:                   0.595
Conditional Recall:                      0.135
Conditional F-measure:                   0.220
Conditional Accuracy:                    0.521
```

Figure 4: MLP classifier results on test set.

The asymmetry in the class counts on the test set is nearly five times as exaggerated as with the dev set. Nonetheless, the model's precision for underperformance predictions is a moderately surprising 59.5%. Noting that the recall is an abysmal 13.5%, we accept that the only foreseeable potential for this model is as a signal to underweight or potentially short a small subset of all possible companies in the S&P 500 index. A portfolio that went short the excess return of these 163 company-quarters would have earned 1.74% under ideal conditions per quarter on average; the long recommendations (with precision 51.1%) would have lost 0.73% per quarter. The best case we can reasonably support is therefore shorting all of the stocks predicted to underperform and hedging with long positions in the index itself (and not the stocks predicted to outperform). This would produce an annualized return of 7.14% assuming no transaction costs with constant reinvestment and rebalancing. Of course, such performance is practically unattainable, much as we suspect is the case for the 10% alpha proposed by Li (2006).

Works Cited

Fama, E. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 383-417.

Li, F. (2006, April 21). Do Stock Market Investors Understand the Risk Sentiment of Corporate Annual Reports?

Sean L. Humpherys, K. C. (2011). Identification of fraudulent financial statements using linguistic credibility analysis. *Decision Support Systems*, 585-594.

Sunita Goel, J. G. (2010). Can Linguistic Predictors Detect Fraudulent Financial Filings? *Journal of Emerging Technologies in Accounting*, 25-46.

Yuh-Jen Chen, C.-H. W.-M.-Y.-K. (2017). Enhancement of fraud detection for narratives in annual reports. *International Journal of Accounting Information Systems*, 32-45.