Lazy Prices: Vector Representations of Financial Disclosures and Market Outperformance

Kuspa Kai
Department of Computer Science
Stanford University
kuspakai@stanford.edu

Victor Cheung
Department of Computer Science
Stanford University
hoche@stanford.edu

Alex Lin
Department of Computer Science
Stanford University
alin719@stanford.edu

Abstract

The "Efficient Market Hypothesis" (EMH) states that market outperformance is impossible through expert selection because each stock price efficiently incorporates and reflects all relevant evaluative information. We study the validity of EMH by analyzing the latent information of financial disclosures year over year. Specifically, we explore the concept of "Lazy Prices", the idea that changes in financial disclosures are correlated with a decrease in market capitalization, using natural language processing methods to factor in these changes the market may not capture. We created a novel database of financial disclosures represented as GloVe vectors from 60,000 raw 10-K documents filed with the Securities and Exchange Commission (SEC) from 1994-2016, and trained several models to predict future market performace. Because our best model did not acheive cross-validated prediction accuracy greater than 56%, our model provides evidence in favor of Efficient Markets. We present our dataset, methodology for latent information mining, and results as well as a discussion of future improvements.

1 Introduction and Related Work

"Lazy Prices" (Cohen, 2010) found that firms that modified their periodic financial reports rather than defaulting to boilerplate tended to perform worse in the future compared to firms that did not modify their disclosures. This indicates the existence of abnormal returns. For example, suppose a company changes their annual 10-K disclosure by inserting a sentence into a section describing risk factors. Knowing which particular risk factor was added is not necessary for evaluating market performance in this case, because the relevent feature is the implicit information that risk has changed. The measures used by Cohen et. al. were TF-IDF and other string edit distances. Cohen et. al. used the magnitude of edit distances between documents as a scale for portfolio mangagement, buying "non-changers" and shorting "changers". Using this method, they acheived a rate of return of 30-60 basis points month over month over the following year.

Of particular interest is the possibility that more sophisticated parsing and representation of documents may better capture latent information of the exact changes that lead to financial outperformance. Finding methods that capture semantic meaning or hierarchical structure in changes to these financial disclosures that are otherwise obscure to the market could plausibly form the basis of a more effective portfolio management strategy. We use neural networks to autonomously learn the relevent differential information contained in consecutive financial filings. This approach has

several advantages over String edit-distance because it can represent complexities in the difference between documents. Once we vectorize document text into a feature space semantic meaning and hierarchical structure may be learned using the neural net.

The success of this strategy depends on the degree to which the Efficient Market Hypothesis is true. It claims, in weaker and stronger forms, that all relevant information governing the value of securities are already incorporated into the price of the security, which is then the best estimate of the value of that security. Fama et. al. notes that the prices of securities will also over-adjust to new intrinsic values as often as they under-adjust, and may adjust prior to new information being made public or after. This would make any investment strategy based on identifying mispricing nearly-impossible, thus invalidating the existence of abnormal returns over the long run.

If Efficient Markets is true, then no amount of abstraction and parsing can consistently predict outperformance. This complicates our model evaluation, since poor performance may either indicate a poor model or that the task is intrinsically impossible; however, if markets are not efficient and the "Lazy Prices" results are reproducible over our data set, then we should be able to achieve good results given good models.

2 Approach

2.1 Data Representation

The Form 10-K is an annual filing that comprehensively describes a company's performance for a fiscal year. All US domestic companies are mandated by the Securities and Exchange Commission (SEC) to file a 10-K each fiscal year.

On the corpus, we train our own word embeddings based on the Word2Vec and Paragraph2Vec paradigms. We have XX tokens with a vocabulary size of YY over ZZ documents in total.

We also use an alternative approach through GLoVe vectors over the Wikipedia corpus. Documents represented through this scheme is the arithmetic average of the GLoVe vector for each of the words in that document. We repeat this for each of the 50, 100, and 300 dimensions available.

Item	Business
Item 1A	Risk Factors
Item 1B	Unresolved Staff Comments
Item 2	Properties
Item 3	Legal Proceedings
Item 5	Market
Item 6	Consolidated Financial Data
Item 7	Management's Discussion and Analysis of Financial Condition and Results of Operations
Item 7A	Quantitative and Qualitative Disclosures about Market Risks
Item 8	Financial Statements
Item 9	Changes in and Disagreements with Accountants on Accounting and Financial Disclosure
Item 9A	Controls and Procedures
Item 9B	Other Information
Item 10	Directors, Executive Officers and Corporate Governance
Item 11	Executive Compensation
Item 12	Security Ownership of Certain Beneficial Owners and Management and Related Stockholder Matters
Item 13	Certain Relationships and Related Transactions, and Director Independence
Item 14	Principal Accounting Fees and Services
Item 15	Exhibits, Financial Statement Schedules Signatures

Figure 1: List of SEC Form 10-K Sections

These representations were developed both at the document and the section level. Our decision to split the data in this fashion was driven by the hypothesis given in the "Lazy Prices" paper. We hypothesized that analyzing changes at the granularity of each section would result in a more meaningful representation of the document semantics. Since sectios many do not experience significant changes year-on-year, a GLoVe-averaged document level representation may have a very low signal-to-noise ratio. Comparing individual sections instead allow us to focus on the semantic differences between smaller components.

Additionally, the "Lazy Prices" paper considered the relative changes between individual sections, specifically noting that some sections, such as Item 7. Management's Discussion and Analysis of Financial Condition and Results of Operations, were on average much more dynamic than others. This section-based representation also allows us to train models only on an individual section. Since many sections do not experience significant changes year-on-year, a even small change in a normally stagnant section could indicate a larger shift in the company's material performance.

2.2 Data Acquisition

We used the SEC Edgar database as our main source for filing information. As of 1994, the SEC has mandated that all companies submit a digital filing of their 10-K forms. These are available in multiple formats - HTML, text, and XBRL (eXtensible Business Reporting Language). Since we are most interested in the textual information, rather than specific descriptions and reports, we focused on acquiring HTML and text documents. SEC EDGAR does not have an API with which we can rest documents. Consequently, downloading the filing documents from SEC Edgar (REFERENCE THIS: https://github.com/rahulrrixe/SEC-Edgar) required the development of a scraping tool. Our scraper was loosely based off of the SECEdgar Python library, but we eventually developed our own expanded scraper to better suit our required functionality.

Our final scraper consumes a list of company stock tickers, and requests listings of filing indices from SEC Edgar. We ingested all available 10-K annual filing documents from all companies listed on the NYSE and NASDAQ exchanges from 1994 to present. The crawler parses the listing year, and identifies the relevant documents to download amongst other attached files, documents, exhibits, etc. It prioritizes HTML documents over text documents to improve our signal to noise ratio, as we can more easily parse and identify edge cases in the HTML form. We then parse the document to extract its individual sections, saving those as well as the entire document, converted to text. Sections were identified by parsing tables and lists of links within the original HTML file.

However, our HTML parser was not able to identify linked sections in all files, and was not at all able to parse .txt-based filings. As such, we developed a second parser to ingest all of our related .txt documents and perform search-based parsing to identify sections. We applied this tool to our HTML files as well, in order to extract any sections that may have previously been missed. In our handling of the downloaded data, we prioritized sections that were parsed directly from HTML, and used the .txt parsed files to augment our data where necessary.

We implemented an error checking layer on our parser that checks for which section IDs were derived from the raw data, matching possible concatenation errors for each section. Each permutation of possible concatenations between sections is considered and giving a distinct section ID, so that training examples only compare macroscopically alike sections between consecutive years while allowing for the word by word differences we sought to capture. This error checking step was vital to our preprocessing because comparing macroscopically disimilar sections from year to year would misrepresent training examples as containing much higher degrees of change than actually contained in the data.

2.3 Data Preprocessing

Once we acquired our dataset, we needed to identify 'valid' pairs of documents to compare, generate their proper embedded representations, and then prepare them as inputs to our neural network. A pair was considered valid if it contained two documents from consecutive years with a matching section ID. Each document was cleaned of any punctuation, numbering, or uppercase lettering. Each word was tokenized, and vectorized using GLoVe representation trained on the Wikipedia datset. To represent a document, we took the mean of each word embedding in the document. This allows us to compare two documents with variable lengths.

The labels were created using data from Bloomberg Historical Market Capitalization, and are denoted as a one or a zero. A label of 1 corresponds to a 10K section whose differences from the previous year's analagous section yields a positive change in market capitalization one year later. A label of 0 denotates a negative change.

3 Model

We attempt a variety of models. We simplify the task of predicting out-performance by calculating the year-on-year percentage change in market capitalization for each company, then partitioning the changes into five categories from 0 to 4, with 0 being the worst performance (decreases in market cap) and 4 being the best performance (large increases in market cap). This abstracts away from predicting stock price alone, since prices may change drastically for reasons entirely unrelated to performance, such as stock splits, reverse stock splits, share repurchasing programs, and so on.

Input Data	Input Dim.	Architecture	Train	Test	Val
All, Median	Dual	20 x 30 x 2	0.5687	0.4914	0.4955
All, Median	Dual	50 x 2	0.6189	0.4979	0.4983
Section 7, Median	Dual	50 x 50	0.7514	0.493	0.5112
All, Median	Dual	50 x 2	0.8853	0.5022	0.5023
All, Sign	Dual	50 x 50 x Dropout x 2	0.5956	0.5307	0.5351
Section 7, Sign	Dual	50 x 50 x 2	0.8206	0.5456	0.5494
Section 1, Sign	Dual	50 x 50 x 2	0.6968	0.5172	0.5382
All, Sign	Dual	10 x 10 x 2	0.513	0.5364	0.548
All, Sign	Dual	200 x 2	0.5439	0.5372	0.537
All, Sign	Dual	5 x 5 x Dropout x 5 x 2	0.5787	0.5407	0.5388
All, Sign	Single	5 x 5 x Dropout x 5 x 2	0.551	0.5371	0.5445

4 Network Architecture and Results

Our most performant network architecture consisted of two fully-connected hidden layers, each with ReLU activations and L2 regularization.

We tried feeding our dense document representations into several types of networks. We tried changing the number of parameters, the number of hidden layers, and hyperparameters. We found certain sections performed better than others when predicting if a market cap change would be positive or negative.

Market Cap adjusted for inflation

Extracting sections

5 Results and Discussion

We see performance of our models slightly above average on the validation set when run with certain hyperparameters. This is an encouraging sign. Given the number of traders and arbitrageurs who seeks to exploit informational inefficiencies in the market, we may have reasonably expected that no model could have picked up the signal hidden amongst the noise.

Our choice of document representations isn't necessarily ideal - we don't see that performance drastically improves with larger representations at the densely connected layer. This may be due to the choice of granularity we have chosen for the comparisons across 10Ks — differences are captured as well through a smaller dense layer as it is through a larger one. As well, the averaged GloVe vector is a naive approach, and

6 Next Steps

For concrete next steps, we would like to train word2Vec and document2Vec embeddings on the larger corpus we have built up over the course of this project. When initially attempted, we were looking at a corupus across 500 companies and slightly less than 10,000 documents with 238 Million tokens in total. That number has expanded drastically since, and makes training purpose-specific word vectors possible. The well-written nature of most 10Ks make them especially amenable to word2Vec training without further cleaning.

We would also like to explore more sophisticated models using recurrent neural networks over entire sections — this approach may better preserve the meaning of documents. Furthermore, an attention mechanism may help us naturally hone in on the parts of the documents that change year on year or that which has significant impact as related to market performance.

Acknowledgements

We would like to thank the CS224n TAs for assisting with the challenges of this project, Richard Socher for contributing to the model design, and Chris Manning for providing the infrastructure without which this project would not have been possible.

References

[1] Cohen, Lauren and Malloy, Christopher J. and Nguyen, Quoc H., Lazy Prices (February 10, 2016). Available at SSRN: https://ssrn.com/abstract=1658471