Predicting FOMC Actions using NLP; Assesing the impact of matrix sparsity and regularization

Jeremy Lao - jjl359
NYU Computer Science

John Reynolds - jr4716
NYU Computer Science

May 13, 2019

Abstract

The Federal Open Market Committee (FOMC) meets throughout the year to set the target Federal Funds rate. The Federal Funds rate is one of the primary monetary policy tools, and it impacts the cost of borrowing money globally. This paper explores ML and NLP methods, examines the impact of matrix sparsity from NLP methods and regularization from ML models, to predict the potential outcome of an upcoming FOMC using past FOMC statements and meeting minutes to train ML and NLP models.

1 Introduction

In this paper we use methods in Natural Language Processing (NLP) and Machine Learning (ML) to predict Federal Open Market Committee (FOMC) rate actions (hold or change) using text from FOMC meeting minutes, Board of Governors speeches, and FOMC post-meeting statements.

For this analysis we used 713 separate documents (rougly 1.6 Million Words) and 148 Fed Decisions for our data set. Employing the bag-of-words models have a high dimensionality, meaning the space is extremely sparse with a large amount of features leading to a sparse feature matrix. In the analysis we conducted some of the feature matricies generated contained over 2 Billion elements with fractions as small as 0.0025 of the elements containing a non-zero value. Our analysis and techniques used were driven by the issue of feature sizes orders of magnitued larger than the label size and the extreme degrees of sparsness in the feature matrix. To make sure the analysis was at least plausible we created a simulator to detect positive or negative sentiment from randomly generated text and found as the number of sentiment words increased the prediction accuracy approached close to 1.0. For the study the analysis hones in on the mix of distributions containing hawkish or dovish(positive) words.

1.1 git repository for project

`https://www.github.com/jrrpanix/ML9`

2 Objective

1. Since its pretty easy to just blindly apply tools to a set of data we wanted to understand the characteristics of the output from the text processing functions (in our case CountVectorizer) in terms of number of features, matrix sparsity, matrix norm that are generated from calls to this function.

2. Assess the effeiveness of Multinomial Niave Bayes and Logistic Lasso applied to problmes where the feature size is orders of magnitues larger than the label size.
3. Determine whether ML and NLP methods can be used to predict FOMC action (no change or a change in the base rate).
4. Because 70% of the actions are "no action", make sure the prediction is not always just "no action" (which would give a 70% accuracy).

3 Methodology

3.1 Data

The time frame of our study covered 2000 to 2018. We primarily utilized three sources for our data, FOMC statements, FOMC meeting minutes and Boarde of Governors speeches. Details on datasources, extraction and scrubbing are left to the appendix to keep the focus on the methods and results. Preparing the data was a large amount of work and we are not aware of any data sets that combine the three types of documents and then map the documents to a Fed Action.

3.2 Feature Modeling and Data Label Mapping

Documents are mapped to the first rate decision on or before the document date at time ti where j = Statement or j = Speech or j = Minutes

$$\text{Document}_{j,ti} \rightarrow \text{Decision}_{ti+k} = \begin{cases} 1, & \text{if Decision = change,} \\ 0, & \text{if Decision = no change} \end{cases}$$

Model 1: Unstacked
The **Unstacked** model maps each document to a feature vector generated by CountVector given an n-gram(n_1, n_2) where $n_1 <= n_2$ and n_i are integers > 0. Given there are approximately 713 documents, this will generate a feature matrix with 713 rows and K columns where each column will be and instance of the word sequence generated by the n-gram.

$$E[D_t i] = f(ngram(n_1, n_2)[\text{Document}_{j,ti-k}])$$

Model 2: Stacked
The **Stacked** model maps the set of documents which occurr after the previous decision date but on or before the next decision date to one rate decision. This has the effect of contatenating the documents between decisions into one large document, but cuts down the number from 713 to 148. where $n_1 <= n_2$ and n_i are integers > 0. Given there are approximately 713 documents, this will generate a feature matrix with 713 rows and K columns where each column will be and instance of the word sequence generated by the n-gram.

$$E[D_{ti}] = f(ngram(n_1, n_2)[\text{Document}_{j1,ti-k1}, Document_{j2,ti-k2}, ...]),\ ti - 1 < ti - ki < ti + 1$$

3.3 Matrix Sizes and Sparsity from CountVector

For this study we used **CountVectorizer** (**TfiDf** can also be used, but it did not alter the conclustions of this study). When an n-gram is applied to corpus combinatorics diticates a large number of features will be generated and for a given feature, most of the column entries will be zero, resulting in extremly sparse matricies. The table below gives a measure of sparsity = $\frac{non-zeroelements}{totalelements}$ and the resulting number of elements in each n-gram generated matrix in the billions. In general stacking reduces the sparsity by a factor of 4 and the matrix

size by a factor of 5. The L1 matrix norm for all of the matricies ranged between 1200 to 1600.

CountVector Matrix Size				
	Unstacked		Stacked	
N-gram	sparsity	size(billions)	sparsity	size(billions)
1:1	0.03628801	0.008561	0.10635940	0.001757
1:2	0.00544312	0.170058	0.02081416	0.036107
1:3	0.00368457	0.439898	0.01508941	0.092869
3:5	0.00238128	0.900721	0.01059293	0.173040
4:7	0.00224699	1.332560	0.00996789	0.271117
5:10	0.00218455	1.995502	0.00972276	0.402041
8:10	0.00213633	1.019811	0.00955894	0.198627
10:15	0.00210643	2.097187	0.00945586	0.398312
20:20	0.00205055	0.347043	0.00931157	0.074799

3.4 Apply ML to Sparse Matricies generated from CountVectorizer

For this problem the large feature size relative to the the number of labels dicitates looking at Bayesian approach or some from of L1 regularization (HW1 problem 2) to push a large number of coefficients to zero.

- Naive Bayes

$$P(A|x_1, x_2, ..., x_n) = p(A)\prod_{i=1}^{n} p(x_i|A)$$

- Logistic Lasso

$$A = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i} + L1$$

3.4.1 Training, Testing and Determining the efficacy of the Models

For this study we randomly sample 75% of the corpus for training and then predict Fed action using the ramining 25% of the data and calculate

- Accuracy
- Precision
- Recall
- F1 Score

Approximately 70% of the time there is no Action so a model that does nothing will have an accuracy of 70%, but a recall of 0. The results are summarized using F1 score.

3.4.2 The impact of regularization coefficient α on logistic regression

For **Logistic Lasso** we are interested in examining the impact of the regularization parameter α on the fraction of non-zero coefficients remaining after regularization is applied and on the impact of the predictive score on test data. The results show the regularization parameter at the extremes produces the worst results and the stacked model has a higher fraction of coefficients eliminated from logistic lasso. The expected result is the larger regularization parameter reduces the fraction of non-zero coefficients.

Comparision of Logistic Lasso varying regularization coefficient α					
		Unstacked		Stacked	
N-gram	α	F1	frac remain	F1	frac remain
10:15	3.333	0.454	0.000040	0.720	0.000016
10:15	2.000	0.575	0.000066	0.897	0.000029
10:15	1.000	0.647	0.000148	0.893	0.000055
10:15	0.500	0.679	0.000291	0.913	0.000094
10:15	0.200	0.708	0.000339	0.913	0.000134
10:15	0.100	0.710	0.000463	0.923	0.000161
10:15	0.050	0.711	0.000619	0.925	0.000206
10:15	0.020	0.715	0.000764	0.923	0.000243
10:15	0.010	0.713	0.001045	0.913	0.000315
10:15	0.002	0.730	0.001659	0.906	0.000517
10:15	0.000	0.669	0.978049	0.669	1.000000

3.4.3 A comparision of Multinomial Naive Bayes to Logistic Regression with extrem sparse Matricies

We noticed that under extreme levels of sparseness (sparseness measure of less than 0.003) from the Matricies generated by CountVectorizer , Logistic Regression and Logistic Lasso with larger regularization parameters would break down, while Multinomial Naive Bayes would produce usable results. Logistic Lasso with $\alpha = 0.002$ seemed to hold up well even under extreme sparsness.

4 Conclusion

Reducing the sparsity of the matricies generated from CountVectorizer by stacking the documents greatly improved the results. We were able to attain maximum test accuracy of 95.9% using a 10:15 n-gram with a recall of 93.4% and an F1 score of 92.5%. Because of the number of features generated can be in the millions, Logistic Lasso with small penatly proved to be the most effective method for reducing the number of coefficients. The fraction of non-zero coefficients remaining after applying logistic lasso ranged between 3e-05 to 1.6e-03. (Compared to .97 with no regularization using logistic regression). Logistic regression ($\alpha > 0.05$) and Logistic Lasso both broke down at extreme levels of sparsity (2.5e-03). In general the Regularization parameter at the extremes produced inferior results. MultiNomial Naive Bayes with Laplace smoothing at zero proved to be the most stable method for producing useful results at the most extreme measures of matrix sparcity. In general no smoothing or large smoothing proved to give the best results with MultiNomial Naive Bayes. Overall Logistic Lasso with $\alpha = 0.002$ produced the most stable results.

A Appendix

A.1 What is the FOMC

The FOMC sets the Federal Funds target range (target rate prior to 2009). The permanent members are the Board of Governors of the Federal Reserve System, President of the Federal Reserve Bank of New York, and the rest of the seats are rotated through the Presidents of the other Federal Reserve Banks. There are twelve Reserve Banks and the Board of Governors oversees the activity, operations, and policies of the Federal Reserve System.

A.2 Why Does the Federal Funds Target Range Matter?

The Federal Funds Target Range is set by the FOMC. The Federal Funds Target Range is the range where the Federal Funds Effective rate (calculated as the volumn weighted median of

eligible transactions reported on FR 2420) is exepected to fix on a daily basis. The Federal Funds rate is the primary monetary policy instrument of the Federal Reserve System, and it influces the level of interest rates domestically and globally. For example, if the Federal Funds effective rate were 5 percent, then the interest rate on a 30 year mortgage would be 5 percent or greater.

A.3 Data

A.3.1 Data Sources

The time frame of our study covered 2000 to 2018. The primary source of our textual data were FOMC statements, FOMC meeting minutes, and Board of Governors speeches from `https://www.federalreserve.gov`. For FOMC statements between 2000 and 2008 we utilized Stanford's FOMC corpus. The Stanford FOMC corpus only has text prior to 2008 and it does not have Board of Governor speeches. The data for the Federal Funds target rate and target range (post 2009) is from FRED St. Louis. FRED offers a wealth of economic data and information to promote economic education and enhance economic research. FRED is updated regularly and allows 24/7 access to regional and national financial and economic data.

A.3.2 Scraping and Text Pre-Processing

We collected data from the Federal Reserve website and pre-processed the data for the document-term matrix.

We utilized Python packages *beautifulsoup4*, *re*, and *urllib* to scrape contextual data from the Federal Reserve website. Our scraping algorithm used regular expressions to handle and remove extraneous *html* and *javascript* text collected by the page scraper. We had to handle non UTF-8 characters upfront and remove them at this stage altogether. We collected over six hundred documents from the Federal Reserve website.

After scraping the data, we pre-processed the text by removing all punctuation, ensuring proper spacing between words, setting all words to lower case, and making all numbers d to reduce the dimensionality document-term matrix. We also used Regex to detect direct references to the Federal Funds target rate/range and transformed those references (a mixture of numbers, punctuation, and the word 'percent') to *ddpercentrate*. Since we were modeling the sentiment of FOMC rate action, this was one of our strategies to directly capture the target rate in the document-term matrix.

This formed the bulk of our text pre-processing and it allowed us to essentially utilize the stored text in the document-term matrix for NLP and ML models.

A.3.3 Meta-Data

Below is a table of the FOMC document metadata:

FOMC Documents					
Document Type	Years	Num Documents	Total Words	Avg Words Per Document	StDev
Minutes	2000-2018	149	738,656	4,957	1,284
Statements	2000-2018	160	62,350	389	185
Speeches	2011-2018	403	915,359	2,271	1,539

Statements are post-meeting communiqué and are generally short, but contains a summary of the committee's reasoning for the decision. Minutes are longer and contain the Staff and Committee member current economic assessments and outlooks. The speeches in the corpus are

from the Board of Governors and may contain clues of their thoughts on their current economic outlook.

A.4 Latent Dirichlet Allocation

We explored latent topic modelling in our research to determine how a machine learning algorithm would classify $k = 3$ topics, a proxy for the Federal Reserve's three objectives.

Given the Federal Reserve's mandates of stable prices, maximum employment, and financial stability, we explored Latent Dirichlet Distribution of the speeches and minutes' word distribution over three latent topics. We were interested in exploring how well a topic model would distribute words over the three objectives and if the meaning of the distributions were relatively clear.

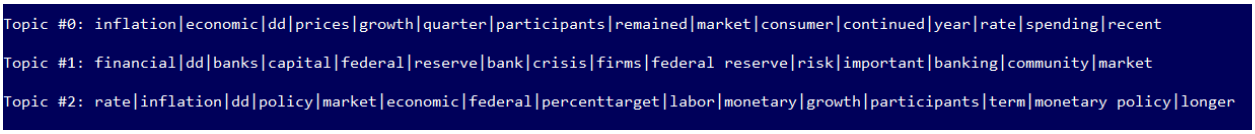

Figure 1: Top 15 Words Distributed Over Three Latent Topics

We found that the topic model generally distributed the speeches and minutes into the three categories fairly well. It was interesting that "Topic 1" referred solely to financial stability, "Topic 0" primarily referred to inflation and spending, and "Topic 2" returned a word distribution that included labor, growth, and lower for longer monetary policy.

References

[1] Hansen,Stephen, McMachon, Michael *Transparency and Deliberation with the FOMC:a Computational Linquistics Approach*, CEPR Discussion Paper, (2014)

[2] Puri, Indira *Using Machine Learning to Predict Interest Rate Changes from Federal Reserve Proceedings*

[3] Ranganath, Rajesh *NYU - CSCI-GA.2565 ML Lecture Slides Spring 2019*