Predicting Legislative Success

Becky Abramowitz
rabram@seas

Jenny Chen
jennche@seas

Graham Mosley
gmosley@seas

Abstract

This document contains an analysis on an algorithmic predictor of the likelihood that a bill becomes a law given textual and contextual information. The report documents the results of experimentation on the features in use and gives the performance of various Machine Learning algorithms on data from the United States Congress since 2013.

1 Introduction

In the past year, thousands of bills have been brought to Congress. Of these, fewer than 6% made it to a vote and fewer than 2% became laws (GovTrack.us, 2018). This discrepancy reveals a large inefficiency in Congress - despite the time and effort that goes into crafting a bill, the overwhelming majority are unlikely to pass. A means of predicting whether a bill will become law could increase Congressional efficiency and focus.

This paper discusses an approach to predicting whether a bill will become a law that utilizes textual and contextual data regarding the bill and Congress. Our best model has a 97.4% accuracy and a .54 f1 score.

2 Problem Definition

2.1 Background on US Legislative Process

The Legislative Branch of the United States government is bicameral, meaning that it is split into two bodies: the House of Representatives and the Senate. The Senate has two senators from each state whereas the House of Representatives from every congressional district. Since there are far more congressional districts than states, members of the American public is more likely to request bills via a representative in the House of Representatives than a Senator. Bills start as ideas brought from citizens to their representatives, who can sponsor the bill and bring it to the rest of congress. The bill is assigned to a committee, and sometimes also a subcommittee, which studies and re-words the bill. If the committee approves and releases the bill, it is put up for debate and vote by the rest of the House of Representatives. If the bill passes by a simple majority, it moves on to the Senate, where it is assigned to another committee and again revised and voted on. A simple majority from the Senate would lead to a joint amendment process, followed by a final vote from the House of Representatives and the Senate. If it passes this vote, it becomes a law unless vetoed by the President in 10 days (of Representatives, 2018).

2.2 Data Used

The data used in this report comes from several repositories that pull from data directly published by the United States Congress. These data sets are the known as a primary source for legislative updates, and are used and processed by gubernatorial analytical tools such as GovTrack.

There are two primary datasets in these repositories: bill-specific information and congressional contextual data. The bill-specific information includes over 10GB of pure bill text tracing back to the beginning of the United States government. Due to the vast size of the data set and limited performance the Congressional data parser, only data from the 113th (2013-15), 114th (2015-17), and 115th (2017-Present) Congresses were used for this report. Since the 115th Congress is not yet complete, the labels for these bills cannot be tested with, and thus the bills can only be used for prediction.

The **bill dataset** used had over 30,000 bills (@UnitedStates, 2018) and includes information such as:

1. Raw bill text

2. Metadata on the bill (e.g. status, introduction time, sponsors/cosponsors, summary, potentially the subject of the bill)

The **context dataset(s)** contain information on Congress, with information on:

1. Current and historical legislators (e.g. demographics, party alignment, senate or house)

2. President and vice president

3. Committees and subcommitees for both Senate and House

4. Dates of the sessions of Congress

These data sets were used to train three models: a text-only model, a meta-data and context-only model, and a combined model. The text-only model uses BoW, CoW, and TF-IDF for classifying the verbiage of the bill. The context model uses the committee(s) and subcommittees assigned to the bill, the senator(s) or representative(s) sponsoring the bill, the net political association of Congress, and the party of the Executive branch to define the feature space. The combined model synthesizes both datasets into a combined feature space and operates on those units.

2.2.1 Feature Exploration

Feature selection and extraction was a non-trivial component to this problem. The data stored in the repositories is of varying types (txt, csv, json, tsv) and is not always complete. For example, some bills have been classified into a multitude of subjects while some are not classified at all, and not all legislators follow the standard term length. The data itself was sometimes not consistent (e.g. regarding term length/dates). Since the data set is so large and sparsely spread across various files, it is important to be selective in the features used as well as to properly parse them.

We matched legislators to sessions by parsing out their term dates and cross fitting them to see where they fit. A similar process was applied to presidents and vice presidents as well. Committees were also parsed from their original format into a mapping that made it easy to vectorize and use as a feature vector.

We can see here that most bills seem to end up in the referred category, or go to committee. This indicates that committee is likely an important feature space. We also see particular members of congress are much more likely to sponsor a bill, which is another feature that could be useful.

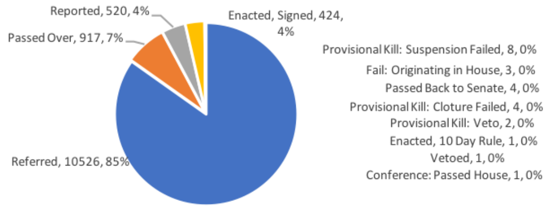

Figure 1: Where do bills end up?

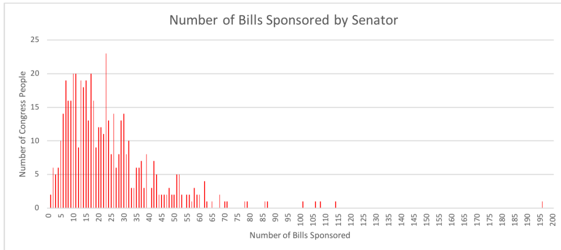

Figure 2: How many bills do people sponsor?

2.3 Modeling Approach

The goal of this algorithm is to predict an output of whether or not a bill will become a law. This is a purely binary classification approach. However, due to the overwhelming number of bills that do not progress to legislature, an always-false classifier can produce an accuracy around 97%, it is more interesting to potentially also know the likelihood a bill will pass versus a simple binary classification. Thus, we utilized both binary classification and regression algorithms in this project. In the regression, thresholds were set to determine the probability at which it would be predicted that a bill would become legislature. The intended use of this predictor would be to be able to pass it a bill in its initial draft stage and predict whether or not it will become a law, helping Congress or observers determine which bills to prioritize.

It can be expected that there is some pattern to the features that cause bills become laws, and thus the algorithm will have an accuracy greater than an always-false classifier. It can also be expected that some of the features will be more influential than others, and thus altering the feature space - through both augmentation and reduction - will enhance the model's performance.

Our approach was to run a text-only classifier, a meta-data/context-only classifier, and a combined classifer and compare the results between them. The exact features used for each is described under section 2.2.

2.4 Algorithms Used

We choose to focus on models that either had a binary classification, or ideally were continuous in order to predict a probabilitity. In total, we utilized these algorithms:

- Support Vector Machine
- Random Forest
- Logistic Regression using L1 and L2 regularization

Since this a highly imbalanced set of classes, we decided to use an ”always false” classifier as our baseline.

3 Evaluation

3.1 Evaluation Metrics

The resulting algorithms are evaluated on a variety of metrics. Accuracy, while utilized in this context, is not a particularly useful metric of performance, as an always-false classification tool would yield almost 97% accuracy. Therefore, it is more indicative to utilize metrics that are dependent on the number of actual and predicted positive values, such as precision ($\frac{TP}{TP+FP}$), recall($\frac{TP}{TP+FN}$), and F1 Score (weighed average of precision and recall). This report also uses AUC (Area Under the Curve) in an ROC (Receiver Operator Characteristic) curve, which is helpful since it looks only at positive outputs, and whether they were true or false, as well as the Brier Score, which is mean squared error. All of these metrics have an optimal value of 1 except Brier Score which has an optimal value of 0. We saw these metrics vary between the different models, which is a good sign they are picking up on differences on how well the classifiers are doing.

The data used in training is from the 113th Congress, and testing is done on the 114th Congresses. The 115th Congress is not good for testing or training data because it has not been completed, but the model is still capable of making predictions on the 115th Congress dataset. The data used regarding each bill in all Congresses is limited to the information on the first draft of the bill, making the model more capable of handling prediction from initial proposition.

3.2 Results

There are three feature sets used and analyzed in this report: Text Only, Meta-Data Only, and Combined Text with Meta-Data. The results of each of these feature sets are described and analyzed below.

Code available upon request

3.3 Summary of Best Models

Data Used	Model	Accuracy	F1 Score
Context only	LogReg	97.1	.532
Text only	LogReg	96.8	.295
Combined	LogReg	97.4	.543
Baseline	False	96.9	0

3.3.1 Text Only Classification

Textual features were obtained from the text of the original bill when it was first introduced into the House or Senate. This allows us to make a prediction on the passage of the bill immediately when it’s introduced. We experimented with Bag of Words (BoW), Count of Words (CoW) and TFIDF of ngrams from the bill text and title as well as the length in words and characters of the bill. We found that using logistic regression with the TFIDF of all ngrams in the range of 1-4 generated from the official titles of the bill worked best. Most of the official titles were quite long and acted as an unofficial summary of the bill (something that wasn’t usually added to a bill until after it had already passed committee).

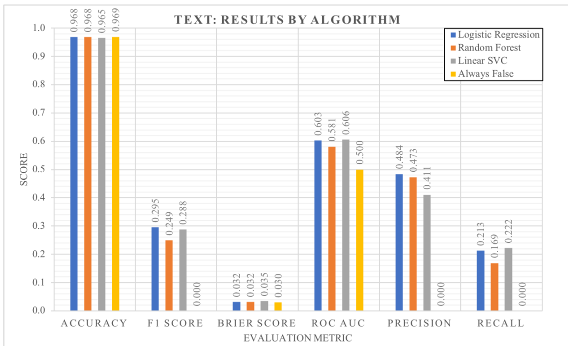

Figure 3: Results from Text only Classifiers

3.3.2 Meta-Data Only Classification

The meta-data classification was performed with varying algorithms on varying feature spaces. This allows for the analysis of what external features are most likely to determine whether a bill becomes a law, and sheds some insight onto the process itself. At its most, the feature space contains the following, which form a 910-feature vector: (1) a binary vector of committees involved with the bill, (2) a binary vector of whether each representative/senator is a member of the bills committee, (3) the history of the bill, (4) the political party of the president, (5) number of committees assigned to that bill, (6) number of representatives/senators assigned to that bill, (7) majority of the House/Senate. Looking at these features as classifiers individually and applying the same standard Logistic Regression algorithm yielded the results shown in Figure 4.

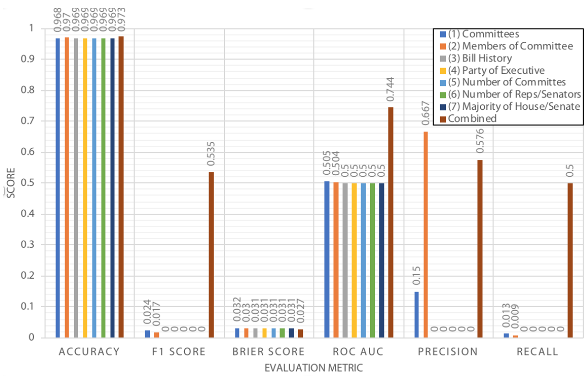

Figure 4: Meta-Data, One Feature Type

From this chart, it is clear that the individual features are nowhere near as effective in prediction as the collective entity. It is also clear from this figure that the most useful features pertain to the committees.

Given that the aforementioned collection of features chosen is effective together, as shown above, the algorithms for classification can be analyzed. Figure 5 shows the results from optimized versions of the three aforementioned algorithms - Logistic Regression, Decision Trees, and SVM - as well as the baseline classifier.

As shown in Figure 5, Logistic Regression was the best scoring algorithm on all of the metrics. It is interesting to note that in the tuning of the Logistic Regression and SVM constants, the models performed significantly better with a larger C value, meaning that the results were less regularized. This discrepancy can be justified with the class-imbalance, as the regularization causes an increasing number of instances to be predicted with a false label.

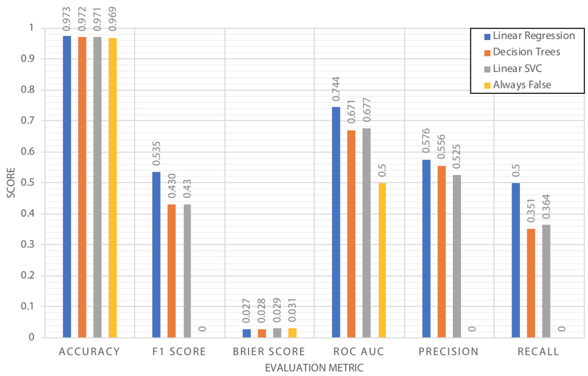

Figure 5: Meta-Data, Varying Algorithms

3.3.3 Combined Text and Meta-Data Classification

For our final classifier, we combined both the text and meta-data features together. We hoped that this would significantly increase our model performance. However, the combined classifier did not seem to score much better than the meta-data only classifier. We also attempted to doing a ”voting” scheme instead, where the text-only classifier was given less weight on its prediction versus the context-only classifier.

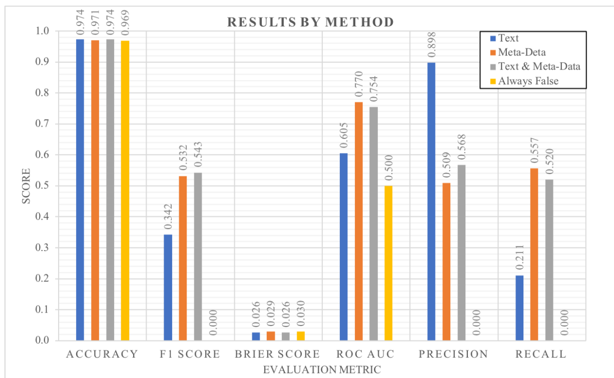

Figure 6: Meta-Data, Varying Algorithms

This graph shows the best results by type of data used, and we can see that the context-only one seems to outperform the others.

3.4 Discussion

Our hypothesis that there are predicting features in a bill was supported. Clearly, both using the text of the bill and/or the metadata and context around it can help you predict how well a bill will do. Perhaps what is a bit more surprising is that the context only model performed almost just as well as the combined model and in some metrics, slightly outperformed it. The text of the bill, logically, contains significantly more information about what the bill is and will do, but you do not actually need the text in order to determine whether or not the bill will succeed.

Because of the large class imbalance, we suspected relatively straightforward, probabilistic algorithms like Logistic Regression would perform the best. We attempted some ensemble methods like Random Forest, but did not necessarily see a massive improvement. If we wanted to further address the issue of class imbalance, we could have upsampled or downsampled to make sure our training and test set had the appropriate data points. It is also possible that using an ensemble "voting" method that combined the two classifiers with different weights and further tuning it may also work better.

3.5 Data Analysis

We investigated both false negatives and false positives but were unable to find any trends in the bills that were misclassified. We believe this is because the legislative process is very complex and bill passage depends on thousands of different features, and so it is likely better to tune the features chosen.

This does illustrate a limitation of this particular project – so many factors go into creating a bill (e.g. political history, conflict of interest, sentiments/personal beliefs, ideology, changes over time) beyond just the text that it is theoretically infeasible to try all of them. So while we have a lot of data points when we just look at the dataset we were given, it is also certainly possible to continuously incorporate more. However, parsimony of the model is also important and so we selected features that we believed to have the greatest weight and tell the most direct story.

3.5.1 Predictions on 115th Congress

Based on our model, we predict that for the 115th congress, of the 8265 bills currently proposed or processed, only about 183 of them will be passed. Number wise, this is quite a bit lower than the last two sessions of congress (which have been about 300), but in terms of percentage is roughly the same passed.

We also ran the model on the tax bill (Tax Cuts and Jobs Act of 2017) that was passed and hailed both as a reform of the tax code but also received a lot of criticism. Our model successfully predicted the bill as passing with a 53% chance, which also reflects the division that the bill caused. Similarly, testing it on HR5087, the Assault Weapons Ban bill, showed a low likelihood of passing and becoming a law.

Based on this, it seems our models do fairly well predicting the likelihood a bill will become a law and in capturing the features that help determine that.

4 Related Work

There has been work done in this space, especially due to the immensely low rate that bills pass. For example, the data sources we used are run and supported by GovTrack, which is focused on tracking the actions of congress. They also do provide some prognosis on how a bill will do in congress, but do not provide any of the source code. We were interested in seeing if it would be possible to replicate some of their results.

GovTrack cited a paper by Yano, Smith, and Wilkerson, who looked into textual predictors about the success of a bill (Yano et al., 2012). Obviously in this project, we also examined metadata as well as the text. There were also some projects at a Stanford course doing a similar thing as us, but again only using the text and looking at the roll call of specific people in Congress (Goldblatt and O'Neil, 2012).

While it is likely the GovTrack predictor uses context rather than just text of the bill, most of the papers we found either did not publish their models or limited their features to only text.

5 Future Work

The most telling future work would be to run the algorithm on bills as they are proposed to Congress to see how the algorithm performs going forward. With further processing and download speed (it took 3 days to download and parse the bill text for the 113-115th Congresses), it would be interesting to incorporate data from before the 113th Congress, since the makeup of Congress is likely vastly different then. We would be able to look into factors that could be more definitive, say for example if Democrat majority legislatures are any different than Republican majority legislatures.

Another enhancement would be to look at the stages of the legislative process and predict how far a particular bill will make it - for example, whether a bill makes it out of committee, or if it is likely to make it to the Senate but get rejected by the Senate, or how likely a bill is to get a presidential veto. Additional datasets that could be included are FEC fundraising amounts, recent news headlines, and/or analysis on social media posts by members of congress or other influential political members. This may help capture even more of the "context" around a particular bill that influences how likely it is to succeed.

6 Conclusion

The legislative processes on Capitol Hill is complex and requires both time and effort to navigate successfully. The low rate that bills pass and become law is not an anomaly – from 2001 to 2015, 70,000 bills were introduced but only 2,513 were enacted. Here, we model the likelihood of a bill becoming a law through various models and datasets while still carefully accounting for the class imbalance. Ultimately, we found that a model that utilizes only the context surrounding the bill performs better than a model trained on just the text of the bill. This indicates the important features to the enactment of a bill are related to factors surrounding the bill, such as who sponsors the bill, the political makeup of congress, and which committee the bill is referred to.

Hopefully, this can help observers focus their attention on which bills are more likely to pass (or not!) and/or help show patterns in the legislative process that may not be obvious otherwise.

References

David Goldblatt and Tyler O'Neil. How a bill becomes a law - predicting votes from legislation text [online]. 2012.

GovTrack.us. Statistics and historical comparison [online]. 2018.

United States House of Representatives. The legislative process [online]. 2018.

@UnitedStates. United states congress [online]. 2018.

Tao Yano, Noah Smith, and John D. Wilkerson. Textual predictors of bill survival in congressional committees [online]. 2012.