Solomon

An Algorithm Predicting the Survival of Bills
In the House of Representatives

CS 229 Final Project Report
SUIDs: [bjang1, dokwon, gyujinoh, jipark21]
Authors: [Brian Jang, Do Kwon, Gyujin Oh, Ji Park]

1. Introduction

Predicting the survival of a congressional bill is analogous to the general problem of predicting human responses to general ideas. A Congressional bill is a collection of long-living social themes, such as abortion, jobs, and economy. By looking at the voting records of Representatives on bills of the same subject, we can predict how they will vote on future bills.

2. Prior Work

While there has been previous work on predicting the survival of bills in Congress, we have yet to discover algorithms that attempt to capture the topicality of the bill text. Two projects have aimed for bill survival prediction: the first by Cain et al. in a CS 229 project, and Yano et al. While Yano et al.'s work focused on Congressional committees, Cain et al.'s work only focused on the textual similarity of the bills without capturing the topicality. Previous work, by neglecting to account for the topic of the bills, fares much more poorly than our algorithm.

3. Task Definition

We divide the task into three components: topicality analysis, individual vote prediction, and party vote prediction. First, using the pLSA algorithm (Probabilistic Latent Semantic Analysis) we look at the text of the bill to determine the topicality. Second, given a new bill, we try to predict the way each representative will vote, given his/her voting record on bills with a similar topic structure. Since bills have a binary decision structure (one either chooses to support the bill or oppose it) we formulate the problem as a logistic regression problem.

4. Method

4.1 Intuition

When presented with a proposition, (in the form of a bill in the Congressional context) human beings make decisions on the basis of its related themes. Thematic ideas are the key basis underlying human decision—we decide we agree or disagree with the ideas of capital punishment, abortion, and free speech. The key intuition of our algorithm is capturing this "topicality" behind the human decision-making process.

4.2 Preprocessing of Data

We collected the full text of bills tendered to the House of the 109th Congress (2005 ~ 2006), as well as the votes of each Representative on the bills from the US Government Printing Office.

(http://www.gpo.gov/fdsys/) Each bill has an associated metadata that is already tagged with its relevant topics. We parsed the full text body of the bills, its metadata, and the votes of each representative, and created xml files that suit our purposes.

Before the text of the bills could be used, we had to perform a substantial amount of preprocessing. To remove extraneous data, we stemmed each of the tokens, removed punctuation and stop words, and converted all words to lowercase.

4.3 Leveraging Topicality

One of the most important features of a bill responsible for each person's decision would be the list of topics relevant to it. Models that aim to draw topics from documents are called *topic models*. Among various topic models available, we in particular were interested in topic models that can take advantage of a "pre-assigned range of topics." To be more specific, in our settings, every bill has several tags associated with it, and each tag indicates a specific topic that might be relevant to the bill. In this sense, chose the Probabilistic Latent Semantic Analysis (or, pLSA), formulated as follows:

$$P(w,d) = \sum_c P(c)P(d|c)P(w|c) = P(d)\sum_c P(c|d)P(w|c)$$

Figure 1. The Probabilistic Latent Semantic Analysis algorithm (Source: Wikipedia)
In the above equation, w is a word, d is a document (in our setting, a bill), and c is a topic.

The advantage of using probabilistic model is that we can lay additional conditions that $P(c|d) = 0$ for every topic c not occurring as a tag of d. Given such conditions, like SVM, there are very few nonzero summand, which enables us to handle a larger number of topics with faster speed. Also, among many ways to maximizing $P(w, d)$ in pLSA model, Expectation Maximization(E-M) method preserves the condition $P(c|d) = 0$ in each step. In this spirit, we used E-M method to implement pLSA.

$$\begin{aligned} P(z|w,d) &= \frac{P(w,z,d)}{P(w,d)} \\ &= \frac{P(w|z)P(z|d)P(d)}{\sum_z P(w|z)P(z|d)P(d)} \\ &= \frac{P(w|z)P(z|d)}{\sum_z P(w|z)P(z|d)} \end{aligned}$$

Figure 2. E-step of EM-method update rule (Source : A Tutorial on PLSA, http://arxiv.org/abs/1212.3900)

$$\begin{aligned} P(d) &= \frac{\sum_w \sum_z n(d,w)P(z|w,d)}{\sum_d \sum_w \sum_z n(d,w)P(z|w,d)} \\ &= \frac{n(d)}{\sum_d n(d)} \\ P(w|z) &= \frac{\sum_d n(d,w)P(z|w,d)}{\sum_w \sum_d n(d,w)P(z|w,d)} \\ P(z|d) &= \frac{\sum_w n(d,w)P(z|w,d)}{\sum_z \sum_w n(d,w)P(z|w,d)} \\ &= \frac{\sum_w n(d,w)P(z|w,d)}{n(d)} \end{aligned}$$

Figure 3: M-step of EM-method update rule (Source : A Tutorial on PLSA, http://arxiv.org/abs/1212.3900)

Note that what we are really interested in is the relevance of each tag (or, topic) to the bill. Thus, after optimizing all the prior and conditional probabilities, we will primarily utilize P($c|d$), the conditional probability of each topic c in a document (or, a bill) d.

4.4 Classification

Given that we obtained the data about relevance of each topic tagged to the bill, we can hone the feature vector of bills more accurately. The downside of using binary classification algorithm only based on the list of tagged topics is that not-so-relevant topics might be too effective on determining the classifier. By setting each entry of feature vector to be P($c|d$), instead of 1 or 0 indicating whether the topic is tagged in the bill or not, the effect of less relevant topics will subside. Indeed, we observed better performance by using detailed conditional probabilities obtained from pLSA step above. Also, to naively include the so-called "party line," we used, as a feature of a bill, which party suggested the bill. Note that the party line also would rather be expressed as a real number, not 0 or 1.

In order to find the optimal classification algorithm, we deployed Logistic Regression, Support Vector Machine, Nearest Neighbor (with 2 neighbors), Decision Tree, and Linear Discriminant Analysis through cross-validation.

5. Result

For accurate evaluation of the algorithms applied, we have randomly selected 70 percent of the data set as a training set and rest of the 30 percent as a test set.

First off, we compare our results incorporating topicality with the results of a previous CS 229 project that attempted to do the same through a simplistic textual similarity model. [2] Incidentally, the previous team used the same dataset and performed near-identical preprocessing—meaning that any improvements must come from our topic analysis. Our algorithm, with little special honing done besides capturing topicality, far outperforms the results of the previous work.

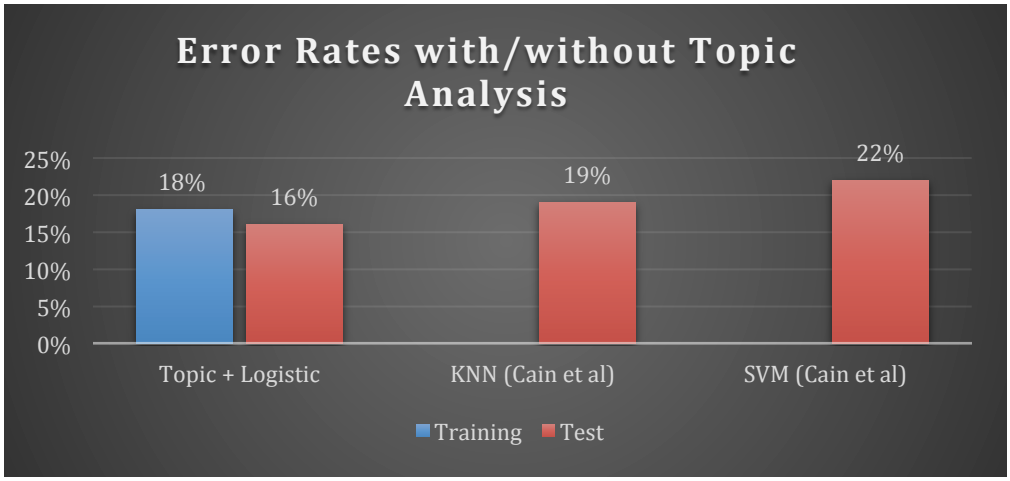

Figure 3. Comparison of error of topic / textual similarity models

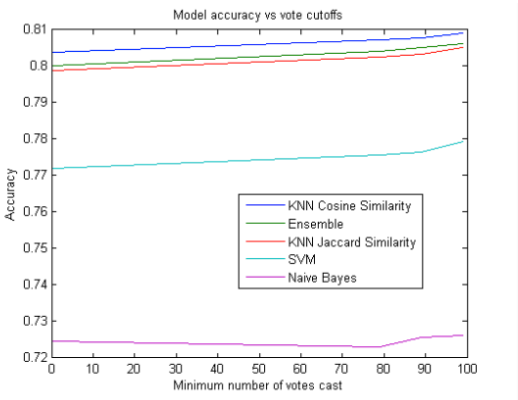

Figure 3: Main plot of accuracies for different models implemented

Figure 4. "Predicting Congressional Outcomes" Results[2]

We also experimented with various binary classifiers, and selected logistic regression, as it was the most performant classifier. We purposefully kept the binary classification simple, as to strengthen the argument our good performance originated from the topical analysis portion of the algorithm.

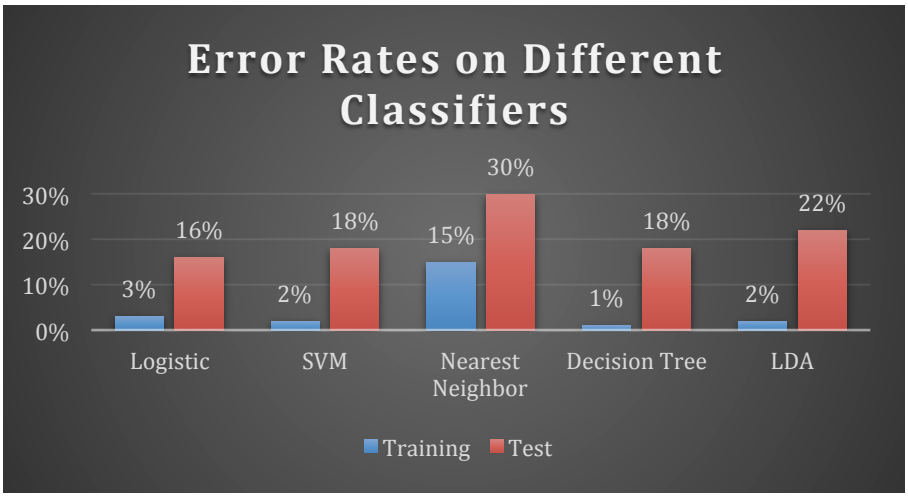

Figure 5. E-step of EM-method update rule (Source : A Tutorial on PLSA, http://arxiv.org/abs/1212.3900)

6. Evaluation

The baseline of our algorithm is a simple logistic regression algorithm without topic analysis, and the results from our implementation are as follows.

Evaluation of Logistic Regression + pLSA: With pLSA, we were able to generalize the error rate; compared to Logistic Regression only mechanism, we were able to achieve lower variance through training error of 16% and test error of 18%.

Evaluation of Best Performant Other Work : Using KNN, Cain et al. was able to get a 19% test error, 1% more than ours.

On the Democratic side, we achieved nearly zero error on "yes" votes on the bill, whereas for the Republicans similarly low error was seen for "no" votes. These results are in line with the then

political reality of a Democrat-controlled Congress with the former party pushing through most legislation, and the latter party investing most party line resources to opposing and filibustering legislation.

The oracle for the algorithm is not clear; as we are trying to predict what human Representatives would do, there is no way to build a predictive model with a zero error rate. Lacking such an objective oracle, we will set the upper bound performance of our algorithm to be one with a zero error, though we will never actually get there.

7. Example

In 2005, the House of the 109th Congress had the vote of ID 'h100', and it was on the bill 'hres202'. Tags on this bill are given as "Congress", "House rules and procedure", and "Taxation". The summary of this bill is given as "Sets forth the rule for consideration of the bill (H.R. 8) to make the repeal of the estate tax permanent." (To give a concrete example, we provide one with very short summary, but for other bills, the summary can be long.)

Based on these tags and summary text, we run pLSA algorithm to weight each subject. With the sparse feature vector utilizing those weights obtained by this algorithm, we train by logistic regression for each representative with the result whether he voted supports or not for this bill. For example, the representative of ID "400004" voted "Yes" for this bill, so his tendency toward voting "Yes" for tags "Congress", "House rules and procedure", and "Taxation" would increase by this training. More concretely, by pLSA algorithm, weight on "Taxation" would be larger than those on "Congress" and "House rules and procedure" for this bill and it affects to the tendency of the representative more.

After training each representative with a large enough history of his votes, we test for testing set with same method: weight on each subjects of the bill, create the sparse feature vector, and predict the result with learned weight vector.

8. Conclusion

Throughout the development and analysis of the Solomon algorithm, we were able to discover the critical role topics play in human decision-making and the computer-powered predictions that try to emulate it. Topic analysis in regular, formatted text such as legislation just begins to scratch at the surface of a much larger, more interesting family of problems: namely, isolating *ideas* and *intentions* from natural language. This project will be the basecamp for further research that tackles those problems, and tackling the challenges of machine comprehension of human language semantics.

9. References

[1] Yano, T., Smith, N.A. & Wilkerson, J.D. (2012) Textual Predictors of Bill Survival in Congressional Committees.

[2] Cain, Z, N.A. & Chua, P, N.A.Gampong, K, N.A. (2013) Predicting Congressional Bill Outcomes.