Predicting Legislators' Votes on the Government Shutdown using Twitter

Iulia Cioroianu, LSE/NYU; Jonathan Nagler, NYU; Joshua Tucker, NYU; Duncan Penfold-Brown, NYU; Jonathan Ronen, NYU; Pablo Barbera, NYU; John Jost, NYU; Richard Bonneau, NYU

Abstract

Can the voting behavior of legislators be predicted based on their social media posts? This paper addresses this question in the context of the October 2013 US Federal Government shutdown, a topic widely debated by legislators on social media and on which we have variation in vote within the Republican party. We test if the language used by republicans who voted yes (Yes-Republicans) and helped democrats end the shutdown differs from that of republicans who voted against the bill that ended the government shutdown (No-Republicans). Using supervised classification algorithms, we show that the voting behavior of Yes-Republicans can be predicted based on the text of their tweets in the period prior to the shutdown. Taking social media data into account also improves the fit of classical models of legislative vote which rely solely on other covariates, such as legislator and district ideology. Finally, using a K-means clustering algorithm we show that based solely on text, we can distinguish between democrats and republicans, as well as between the subgroup of republicans who voted together with the democrats and those who voted against them.

1 Introduction

The debate over the debt ceiling in the US Congress in the fall of 2013 was a highly public and highly partisan legislative debate. The outcome was uncertain, but hinged on the willingness of a subset of Republican legislators to vote in favor of the bill. Eventually 87 House Republicans voted with 198 House Democrats in favor of the continuing resolution that ended the debt crisis. In the Senate, the bill was supported by 27 Republicans. Overall, 114 republican Members of Congress voted with the Democrats.

The legislative stalemate lasted 16 days. During that time, and for the two weeks before the start of the shutdown, members of the House communicated their positions on the ongoing negotiations via Twitter, with members of each party emphasizing different aspects of the debate. Republicans emphasized "#obamacare" and democrats emphasized "#gop-shutdown". There were also differences in the language used by the Republicans who ended up voting with the Democrats, as opposed to those who voted against, the former being less likely to talk about "#obamacare" than the latter, instead emphasizing terms such as "rationale" and "enough".

In this paper we test the hypothesis that legislators reveal how they will vote on upcoming legislation through their tweets. We consider three sets of legislators: Democratic legislators (all of whom voted YES), Republican legislators who voted NO (No-Republicans), and Republican legislators who voted YES (Yes-Republicans). The timing of the vote makes it clear that on the first day of the shutdown there was not a majority available to pass the bill, but by the sixteenth day there was a majority available. If we assume that all Democrats were always publicly willing to vote yes (which the record supports), then we can compare the set of Republicans who ultimately voted yes to those who voted no. The question we would like to answer is: Can we identify the 'switchers' - Republicans who became willing to vote yes during the course of the debate - before the final vote, based on the words in their tweets? We rely on the text of legislator tweets to test if it is possible to distinguish the Yes-Republicans from the No-Republicans, and whether the Yes-Republicans resemble

the Democrats more than they resemble their Republican colleagues.

We start by visually inspecting the language used by each of the three groups (Democrats, Yes-Republicans and No-Republicans), presenting word clouds with the most important words used by each group a month before the end of the shutdown and on the day it ended. We then present the results of a logistic regression which predicts the votes cast on ending the shutdown based solely on classical covariates that we expect to be associated with vote choice by Members of Congress, while setting aside their social media posts. These models will form the baseline against which we will be comparing the performance of our text-based models. We then turn to the text data and present measures of similarity between the three groups, overall and over time. We also analyze the topics present in legislator tweets through a latent dirichlet allocation model. Taking advantage of the fact that our data is already labeled (because we know the outcome of the vote) we use supervised classification to test if Yes-Republicans resemble the Democrats more than they resemble No-Republicans and if classifiers perform better in labeling them as such the closer we get to the end of the government shutdown. We also estimate our classification models by including not only text features but also legislator-specific covariates, such as their DW-Nominate measures. Finally, we check if legislator votes and group membership can be predicted with unsupervised learning, in the absence of information about the actual outcomes, and present results from K-Means clustering models.

2 Background

The 2013 US government shutdown

The series of events that ended with the October 1st 2013 Federal government shutdown started on September 20, when the Republican majority in the House voted to keep government funded conditional on the president agreeing to defund the 2010 Affordable Care Act (also known as Obamacare). The Senate, controlled by the Democrats, refused to compromise and rejected the bill passed by the House. Republicans in the House reduced their

demands to a delay in the implementation of the healthcare law, accompanied by further reductions in spending, but this proposal was also rejected by the Senate, leading to the October 1st shutdown.

Starting on October 10, several attempts to negotiate a bipartisan deal were made. House Speaker John Boehner's attempts were unsuccessful, but Senate Majority Leader, Harry Reid, and Minority Leader, Mitch McConnell, announced on October 16 that their negotiations have been successful, and that a deal to temporarily halt the government shutdown until January 15, 2014 has been made. The proposal passed through both the House and the Senate on the same day and was signed into law by the president that night.

Predicting political behavior using text

Legislative votes

Various forms of text - such as the content of bills and legislative proposals, legislative floor speeches, Supreme Court briefs, opinions and decisions - have been previously used to predict the behavior of political actors.

Several studies use the text of bills or legislative proposals to predict the votes of legislators, often in the context of the US Congress. Gerrish and Blei (2010, 2011, 2012) combine ideal point estimation and topic models to infer the ideological leanings of members of the US House of Representatives and Senate and to predict their votes. Their recent models can also be used to explore the ways in which a lawmaker's voting patterns deviate from party line, depending on the topic of the bill under consideration. Wang et al. (2013) propose a Bayesian model for jointly modeling text and roll call data, which takes into account the temporal evolution in legislator latent features and the spatial location and adjacency of their constituencies. Yano et al. (2012) predict whether bills will survive committee consideration using a baseline set of covariates which have been previously found to be associated with bill survival, and adding information from the text of the bill to this baseline. They find that taking textual features into account improves the model's ability to predict bill survival. Wong et al. (2013) develop a model based on random walks on a heterogeneous graph

which relies on the text of bills, roll call data and political connections between legislators to predict legislative votes.

All of these methods can be used only when roll-call data is available. This makes it difficult to apply them to legislative bodies other than the US Congress, most of which do not keep a record of all roll-calls. To address this issue, Beauchamp (2012) proposes a new Bayesian scaling method and a new vector-based scaling method which uses legislators' written and spoken text, as well as party-membership information to ideologically scale legislators in the absence of roll-calls. Laver and Benoit (2002) locate Irish legislators in the ideological space based on the text of their legislative speeches using Wordscores. Diermeier et al. (2012) use support vector machines to extract the terms that are most informative of conservative and liberal positions in legislative speeches and to predict US senators' ideological positions. Although they do not require roll-calls, these methods rely on the public availability of floor speeches, which are not always recorded in legislatures across the world.

Supreme Court decisions

Text data has also been used in the context of US Supreme Court decisions. Beauchamp (2013) predicts Supreme Court decisions based on the text of Court briefs and oral arguments. The method he proposes uses support vector machines and ensembles of univariate regressions and can predict both decisions by the court as a whole, and individual justices' decisions. Lauderdale and Clark (2012) use votes and opinions from the US Supreme Court and introduce a new approach based on Latent Dirichlet Allocation to to estimate the degree to which those votes are about common issues.

The use of social media text data for political predictions

Social media data has been extensively employed to predict election results or evaluate public opinions towards a certain political actor or topic. Tumasjan et al. (2010) find that the number of party mentioning on Twitter, as well as public sentiment measured in tweet text are associated with party votes in German elections. Bermingham et al. (2011) combine sentiment analysis using supervised learning with volume-based Twitter measures, and show

that social media can be used to predict the results of Irish general elections. O'Connor et al. (2010) also find that sentiment towards political actors measured in social media posts correlates with public opinion polls.

Using the text of social media posts to predict legislative votes

Despite the fact that social media data (in particular Twitter data) has been extensively used to predict election results, no previous studies attempt to predict the behavior of legislators based on the text of their social media posts. However, social media is widely used by legislators across the world, and using this method to predict legislator voter has multiple advantages:

- it does not require roll-call data, which in many countries is either not available or available only for certain types of votes

- it does not require the text of the bill under consideration, which again, depending on the political context, might not be available

- it does not require recorded floor speeches

- it provides a real-time measure of legislator intentionsm, therefore allowing temporal evaluations and comparisons

The 2013 US government shutdown gathered a considerable amount of attention on social media, and US legislators made extensive use of Twitter to talk about the topic. We therefore believe that this event would be a good starting point in our attempt to determine if votes can be predicted using the text of legislators' social media posts.

3 Data and research design

Throughout the analysis we use two types of data:

a) text data - the tweets of Members of Congress for a period of 27 days before the end of the federal government shutdown;

b) legislator-specific covariates, such as their DW-Nominate scores, the votes received in the previous election and whether at they time they were an incumbent or not, and district ideology. Table 1 presents summaries of the measures used throughout the paper.

Table 1: Data summary

	Variable	Details
Text Data	Time period	September 20, 2013 - October 16, 2013
	Tweets considered	14587
	No of MCs who voted	527
	No MCs who tweeted	449
	Tweets/MC	Mean=597, Std.=33.8, Min=0, Max=265
	Tweets/day	Mean=540, Std.=266.2, Min=95, Max=1034
	Tweets/MC/Day	Mean=1.2, Std.=2.4, Min=0, Max=57
	No of Democrats	213
	No of Yes-Republicans	135
	No of No-Republicans	102
	Tweets/group	Dem=6916, Yes-Rep=2935, No-Rep=4736
	Mean tweets/group/day	Dem=256, Yes-Rep=108, No-Rep=175
	Words post-cleaning	3225
	Features post-cleaning	5147
Covariates	DW-Nominate 1	Mean=.08, Std.=.47, Min=-.74, Max=.78
	DW-Nominate 2	Mean=-.07, Std.=.26, Min=-.74, Max=.78
	% votes last election	Mean=64.19, Std.=11.26, Min=35.26, Max=100
	% district vote Obama	Mean=50.89, Std.=15.05, Min=18.5, Max=96.7
	Incumbent	Mean=.67, Std.=.47, Min=0, Max=1

We collected text data for our analysis from Twitter's Streaming API by passing a manually collected set of US Legislator Twitter IDs via the 'follow' parameter. These Twitter IDs represent all the available public Twitter accounts of US Legislators over the period of September 20, 2013 to October 16, 2013 (inclusive). Some legislators have multiple twitter accounts, often for different purposes (for example, some maintain a candidate account, a press account, and a personal account). We collected data from all available accounts. Collected tweets were stored in a MongoDB database. Legislator political party, jurisdiction, nominate-score, and vote information (vote to approve debt plan, whether for, against, non-cast, or not present/not applicable) were collected manually. This political data was matched to twitter account data to link political and vote information to tweets and twitter user characteristics for each voting Legislator. Legislators that did not cast a vote are left out of our analysis.

We compiled the tweets of voting legislators for whom we have twitter data into text documents, rejecting some tweets due to lack of content, and cleaning the textual data of the remaining tweets. We discarded tweets based on the following criteria:

- if they were outside the date range of 9/20/2013, 10/16/2013 EST (inclusive)

- if they were a retweet or modified tweet ("MT") with no additional commentary from the retweeter/MTer

- if they were non-English tweets

- if they were empty tweets (tweets with no textual data, eg: a picture tweet or a tweet containing only a link)

Textual data from tweets was then cleaned as follows:

- removed all tweet entities (embedded objects in tweets, such as links, images, etc)

- hashtags and mentions were retained, though stripped of '#' and '@' characters, respectively.

- normalized whitespace (all whitespace characters converted to single spaces)

- translated common acronyms

- lowercased all characters

- translated common English and Twitter shorthand

- expanded common English contractions

- removed all punctuation

- removed all English stopwords from the NLTK stopwords corpus

- removed all words composed entirely of digits

Depending on the type of analysis undertaken, we then compiled the following sub-documents:

a) aggregate tweets by groups of interest per day (where groups of interest are Democrats, No-Republicans, and Yes-Republicans);

b) aggregate tweets by unique twitter user;

c) aggregate tweets by unique twitter user split into time periods p1 and p2, where p1

= [9/20/2013, 09/30/2013] and p2 = [10/01/2013, 10/16/2013]. These data aggregates will be explained in relation to the analyses they support.

Sources for legislator-specific covariates included the Sunlight Congress API, Poole and Rosenthal's (2014) latest release of DW-Nominate scores, Federal Election Commission election results, and the Project Vote Smart API. A dataset of covariates was built and merged with variables derived from the text data.

4 Results

4.1 Word clouds by group

Word clouds provide an intuitive visual representation of the difference between the choice of words between the three groups. Figure 1 shows the word clouds for pairs of groups, on the first and the last day of the government shutdown, looking at the relative frequency of the terms used by each group. As expected, Democrats and Republicans use different terms. Both at the beginning of the period and at the end, Democrats use words such as "GOP" and "shutdown" whereas Republicans use terms such as "Obamacare" and references to the "Senate". However, when we compare the language of Yes-Republicans to that of No-Republicans in the last two sub-figures we see that there is a strong difference in the choice of words between these two groups. At the beginning of the period, Yes-Republicans use terms such as "rationale", "voting" and "bipartisanship" whereas No-Republicans mention "Obamacare" the "Senate" and "#standuptoreid" more. On the day they decided to change their vote, the choice of terms by the Yes-Republicans does not change much, but Yes-Republicans use less terms that are related to the shutdown and instead talk about "mentalhealh" and "plan", and use the word "enough".

Figure 1: Word clouds - group comparison

4.2 Voting models with covariates only, no text

For now, we ignore the textual data and focus only on other determinants of vote, which we expect to be associated with a higher propensity to vote a certain way. Table 2 presents the exponentiated coefficients from a logistic regression of vote to end the shutdown on the DW-Nominate 1 and 2 scores of the members of Congress (Model 1); as well as the percentage of votes received in the district at the previous elections and a dummy variable for whether they were an incumbent in those elections (Model 2). Models 3 and 4 are similar, but instead of taking into account the votes of all members of Congress, we focus only on Republicans, some of which have voted to end the shutdown.

Table 2: Predicting votes without text

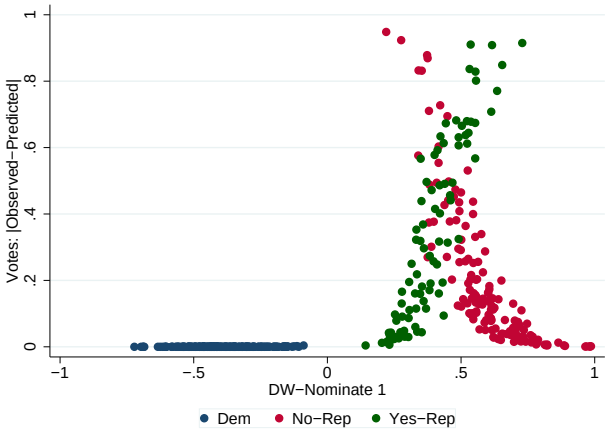
	Model 1	**Model 2**	**Model 3**	**Model 4**
	Odds Ratios (p values)	Odds Ratios (p values)	Odds Ratios (p values)	Odds Ratios (p values)
DW-Nom1	.000016 (.00)	.000013 (.00)	.000017 (.00)	.000016 (.00)
Dw-Nom2	.084 (.00)	.074 (.00)	.084 (.00)	.078 (.00)
% votes election		.993 (.83)		.993 (.83)
% district Obama		1.03 (.427)		1.03 (.427)
Incumbent		.552 (.315)		.553 (.316)
# Obs.	526	440	277	231
LR χ^2	415.34	369.28	141.8	121.41

DW-Nominate 1 and 2 are both significant, the former having the highest impact on the odds of voting in favor of ending the shutdown. The likelihood ratio chi-squares show that all models fit significantly better than a model with no covariates and the differences in BIC and AIC between models 1 and 2, and 3 and 4, are significant and provide evidence in favor of including the other covariates apart from the DW-Nominate scores. As expected, the model restricted to only Republicans performs worse than that which includes the Democrats. Still,

DW-Nominate scores have a strong effect on vote choice.

Figure 2: Difference observed-predicted values by group

(a) All groups



(b) Republicans

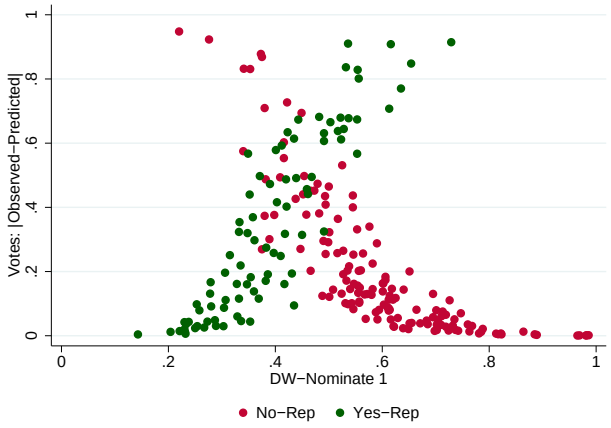


Figure 2 a) plots the absolute difference between the observed vote and the predicted probability of a "yes" vote from the model with all members of Congress and a full set of covariates (Model 2) against the DW-Nominate 1 score, by voting group (Democrats, Yes-Republicans and No-Republicans). We see that predictions are more accurate at extreme values of DW-Nominate. The model predicts the votes of the Democrats perfectly and performs slightly better for No-Republicans than for Yes-Republicans. Figure 2 b) plots similar results using the predicted values from Model 4, in which the focus is on the two subgroups within the Republican party.

Setting a cutoff of .5 for the predicted probabilities from Model 2 (with a full set of covariates), we can compute the classification statistics in Table 3. The ideology of Members of Congress is a strong predictor of vote choice on the shutdown issue. We next examine whether models based on text would perform better than models based on ideology, or whether including information from text would improve the ideology-based models.

Table 3: Logistic regression classification - covariates without text

(a) All groups

		Predicted											
		All			Dem			Yes-Rep			No-Rep		
		No	Yes		No	Yes		No	Yes		No	Yes	
Observed	No	130	26	No	0	0	No	0	0	No	130	13	
	Yes	13	271	Yes	0	209	Yes	26	61	Yes	0	0	

Classification statistics (All)
True positive rate (sensitivity, recall)=91.25%
Positive predicted value (precision)=95.42%
True negative rate (specificity)=90.91%
Negative predicted value=83.33%
Accuracy=91.13%

(b) Republicans only

		Predicted								
		All			Yes-Rep			No-Rep		
		No	Yes		No	Yes		No	Yes	
Observed	No	130	13	No	0	0	No	130	13	
	Yes	13	62	Yes	26	62	Yes	0	0	

Classification statistics (All)
True positive rate (sensitivity, recall)=70.45%
Positive predicted value (precision)=82.67%
True negative rate (specificity)=90.91%
Negative predicted value=83.33%
Accuracy=83.12%

4.3 Analyzing topics: Latent Dirichlet Allocation models

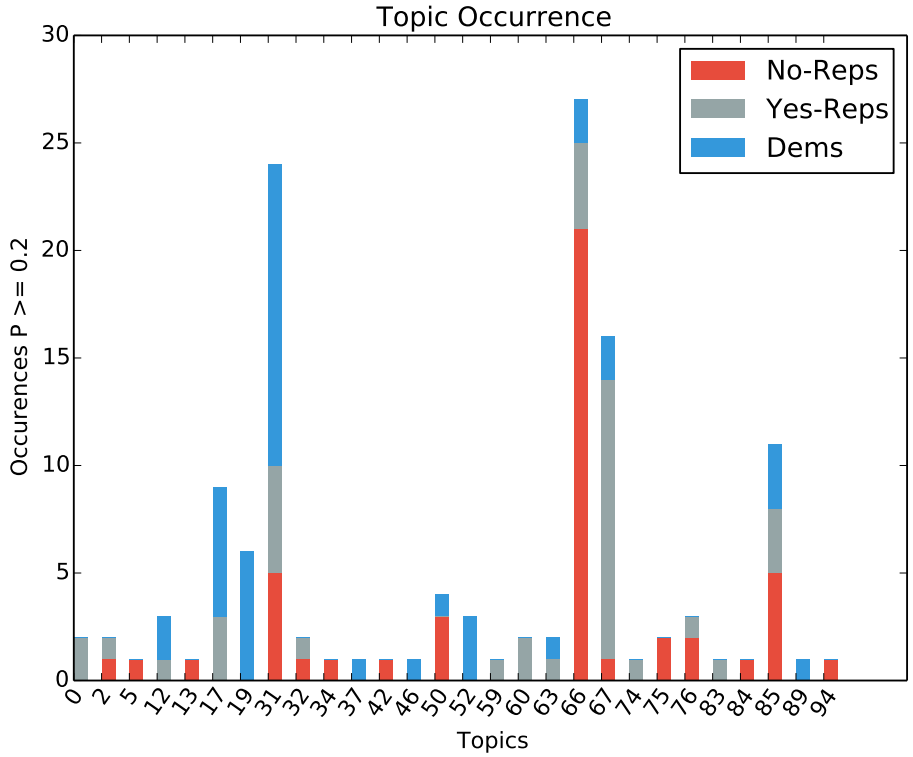
We now turn to the text data. To analyze the topics present in legislators' social media data and to observe the difference in topic space between groups, we used the Python package Gensim to train a latent dirichlet allocation model. Latent dirichlet allocation (Blei et. al, 2003) is a generative probabilistic model that can be used to discover latent dimensions in text data by representing corpus documents as a mixture of k topics, where each topic is characterized by a probability distribution over all words from the corpus.

Our corpus consists of all tweets from each group of interest (Democrats, Yes-Republicans,

and No-Republicans) aggregated by day, resulting in three separate corpus documents for each of the 27 days of our analysis (for 81 total documents). We use a number of topics k=100 (a conservative choice with a stable perplexity score on our corpus), and parametrize the model with a fixed normalized asymmetric 1.0/k alpha prior and a fixed symmetric 1.0/k eta. Our model is trained in batch mode with 20 passes, made possible by our relatively small corpus size, to avoid potential topic instability from online training due to our time-based document construction.

We aggregate data by group and by day in order to understand the progression and presence of topics each group of interest is discussing over time. Summary results are visualized in Figure 3.

Figure 3: LDA models



Observing the topics for each group, it is clear that few, but highly significant topics are shared among all three groups (topics 85, 66, 31), while most are dominated by single

groups or shared between complementary party or voting groups (Yes-Republican and No-Republican, or Yes-Republican and Democrat, respectively). Notably, the topics shared between only Democrats and Yes-voting republicans (topics 12, 17, and 63) are slightly stronger in the corpus than those shared by only Yes-voting and No-voting Republicans (topics 2, 32, 76). It is also apparent that No-voting Republicans are focused more heavily on a single topic (Topic 66) across the range of our analysis, while Democrats and Yes-voting Republicans trade more frequently between topics, potentially indicating a focused strategy by party-line Republicans (in this case, an anti-Obamacare topic).

These divisions give an initial indication that there is separation between voting groups and affinity between Democrats and Yes-voting Republicans in the respective groups' Twitter data.

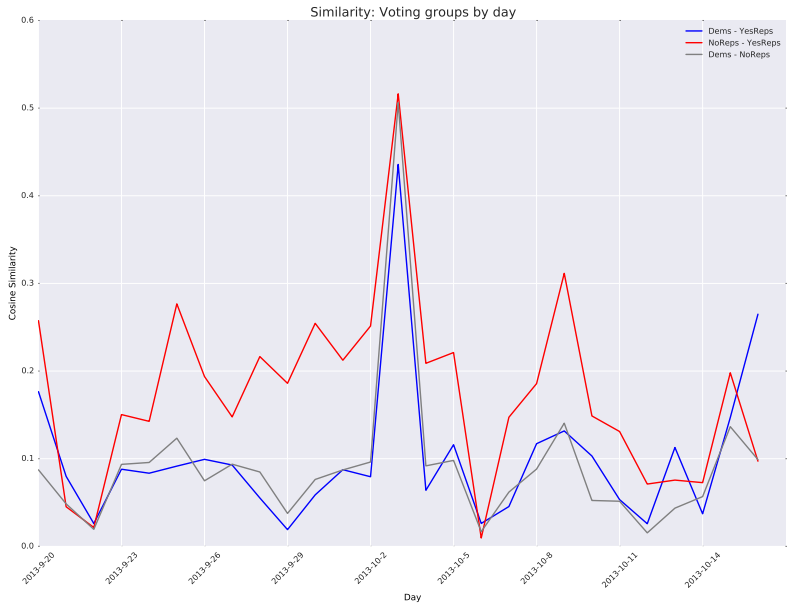
4.4 Cosine similarity between the groups over time

We can quantify text resemblance between the three groups (and over time) by computing the cosine similarity of the vector representations of their corresponding documents. To examine to similarity of texts between groups (Democrats, Yes-Republicans, and No-Republicans), we aggregate all cleaned tweets from each group per day, resulting in 3 groups * 27 days = 81 documents. These documents are represented in two schemes.

First, we use a Tf-Idf (term frequency - inverse document frequency) transformation to represent each document, and consider the cosine similarity between each transformed document for each day. The Tf-Idf transformation represents every document as a vector of weighted word scores. The weighting scheme assigns a high value to terms which occur many times in a small number of documents, and a lower value to terms which occur in many documents. (Manning et al., 2009). Measuring only the magnitude of the vector difference between the document vectors would be inappropriate because we would not be taking relative document length into account. Cosine similarity addresses this issue by calculating the cosine of the angle between vector representations of the documents, therefore providing a measure of document similarity on a normalized space. Figure 4 plots the cosine similarity

between pairs of legislators groups over time, starting on September 20, 2013 and ending on the last day of the government shutdown, October 16.

Figure 4: Cosine similarity pairwise comparisons over time

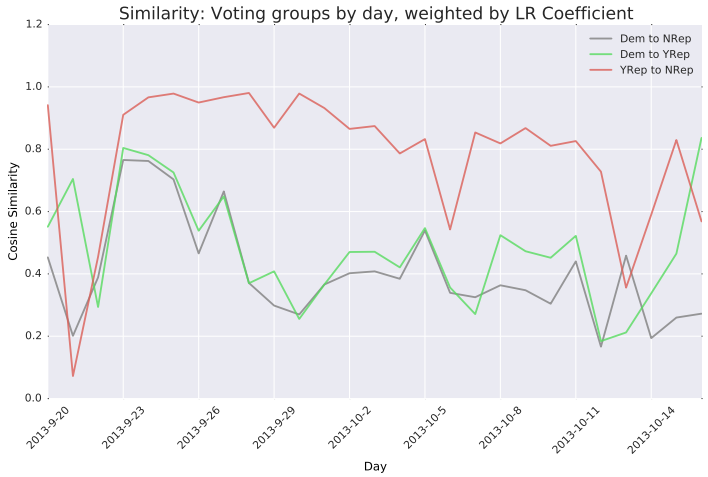


We see the strongest similarity between Yes-Republicans and No-Republicans. We would expect the similarity between Democrats and Yes-Republicans to increase as a we are approaching a consensus. The last three days of our period do show an upward trend in similarity between Democrats and Yes-Republicans, a trend which may indicate a coming together in expressed social media content regarding the impending vote. This trend seems insignificant, however, when observing the fluctuations in the earlier data. Except for these last days, over the entire time frame Yes-Republicans and No-Republicans are more similar among themselves than Yes-Republicans are to the Democrats.

In a second scheme, we create additional documents by aggregating all individual Legislator's tweets. We further clean these documents by eliminating words that appear in fewer than five documents and in more than 95% of documents across the corpus of Legislator documents. This results in 449 legislator documents.

Using a bag of words (BOW) vector representation for each legislator document, in which each word in the corpus is treated as a feature, we train an l2-penalized logistic regression classifier on each document. We use the legislators' votes as the classification target. With the feature coefficient resulting from this model, we weight the initial group-based documents, and plot the resulting cosine similarity of weighted group documents per day (Figure 5). We observe the same upward trend in similarity between Democrats and Yes-voting Republicans in the last three days of our observations, and a reduction in the spikes of similarity before that date. This stability is likely a result of observing the group documents as bags of words as opposed to Tf-Idf documents, thus reducing the impact of relatively unique terms across the whole corpus.

Figure 5: Cosine similarity between the groups by day



4.5 Supervised classification based only on text

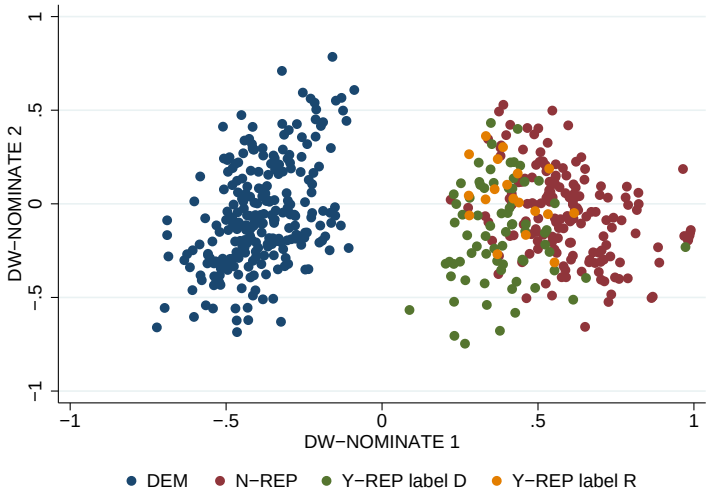
We are interested not only in comparisons between the three groups as a whole, but also in comparing individual members of Congress from these groups, so we now turn to text data organized by legislator. As opposed to using documents composed of tweets from groups per day of observation, we instead aggregate each legislator's cleaned tweets into a single representative document, resulting in 449 documents and comprising a corpus of 18,129

unique words.

Training on Democrats and No-Republicans - predicting group membership for Yes-Republicans

One way to check if Yes-Republicans tweeted more like the Democrats than like No-Republicans is to train a supervised classifier on the texts of these latter two groups and use it to predict the membership of Yes-Republicans to either the democratic or the republican group. A Naive Bayes classifier performs well in the tests against held-out sub-samples of Democrats and No-Republicans (precision=0.81, recall=0.78, F1-score=0.73), so we can use it to label Yes-Republicans as belonging to either one of these groups or the other. The results are somewhat surprising: a very large percentage of the Yes-Republicans (80%) are labeled as Democrats and only 20% of them are labeled as No-Republicans. Figure 6 plots the DW-Nominate 1 and 2 scores of the members of Congress, distinguishing between the Democrats, No-Republicans, Yes-Republicans labeled as Democrats and Yes-Republicans labeled as No-Republicans. It shows that Yes-Republicans who are closer on the DW-Nominate 1 scale to the No-Republicans are indeed predicted to belong to this group based on the text of their tweets, but Yes-Republicans who are closer to the Democrats (and who have indeed voter together with them to end the shutdown) can be identified as such based on just the text of their tweets. This suggests that the text of their social media posts alone could be used to make predictions about the voting behavior of legislators.

Figure 6: Supervised classification - group membership



Training on ALL groups based on vote - predicting vote

Predicting group membership is informative (and in this model specification it overlaps perfectly with vote choice in the training set), but we ultimately aim to predict vote choice for each individual Member of Congress, regardless of their group membership. To investigate whether their Twitter posts indicate a separability in terms of vote direction, we train our classifier on the text of all legislators and make out-of-sample voting predictions on held-out subsets of the data. We test several classifiers common in modeling text data and observe their performance through cross validation. For this part of the analysis we transform our legislator-level text documents into 5,147-feature vectors composed of all unigrams (single words) and bigrams (co-occurring words) that appear in at least five and in less than 95% of all corpus documents. We find that adding the 1,922 filtered bigrams to the initial set of 3,225 unigrams slightly stabilizes and improves classifier performance over three-fold cross validation. Training and classification is performed against a binary target vector representing legislator's vote, with the Yes vote as the positive class.

We train and evaluate the following classifiers on our data:

- L2-penalized logistic regression with C (inverse regularization strength parameter) =

0.01

- Multinomial naive Bayes classifier with prior fitting with smoothing parameter alpha=0.7

- Logistic l2-penalized stochastic gradient descent classifier

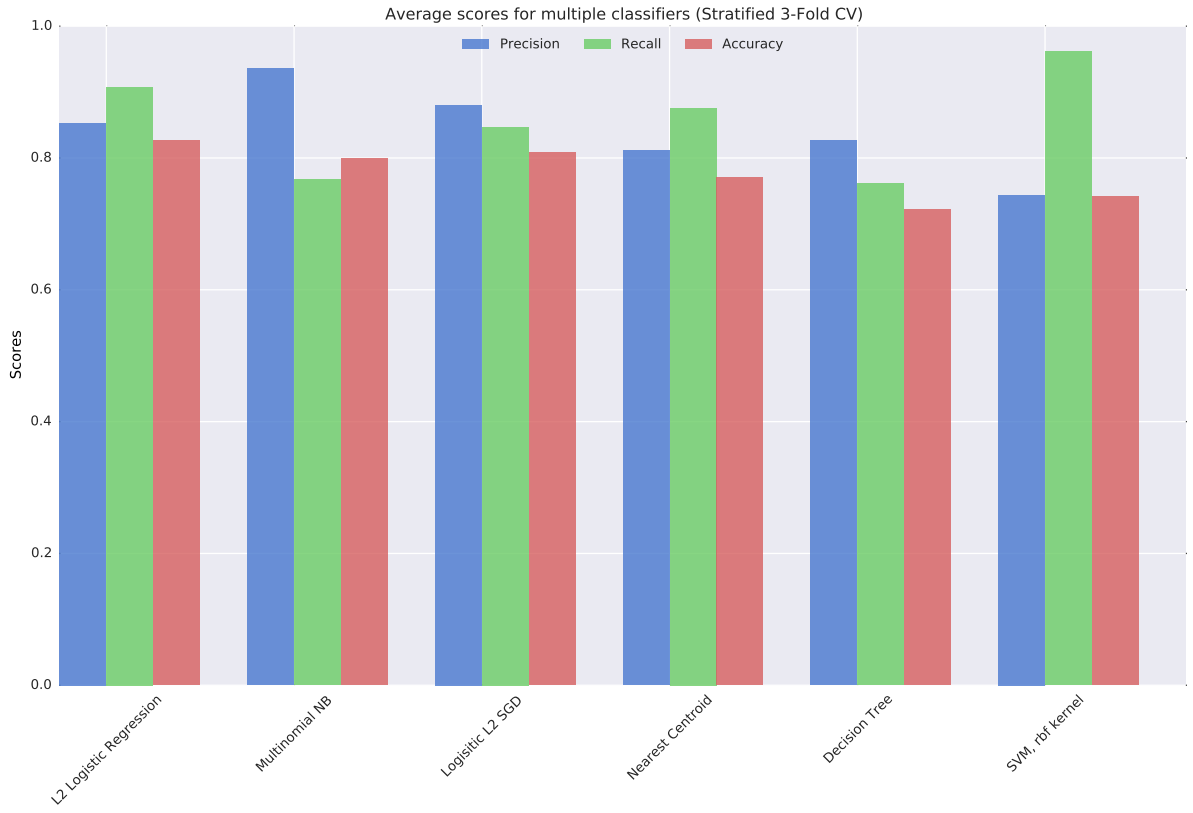
- Euclidean nearest-centroid classifier

- Decision tree classifier

- Rbf-kernel support vector machine classifier

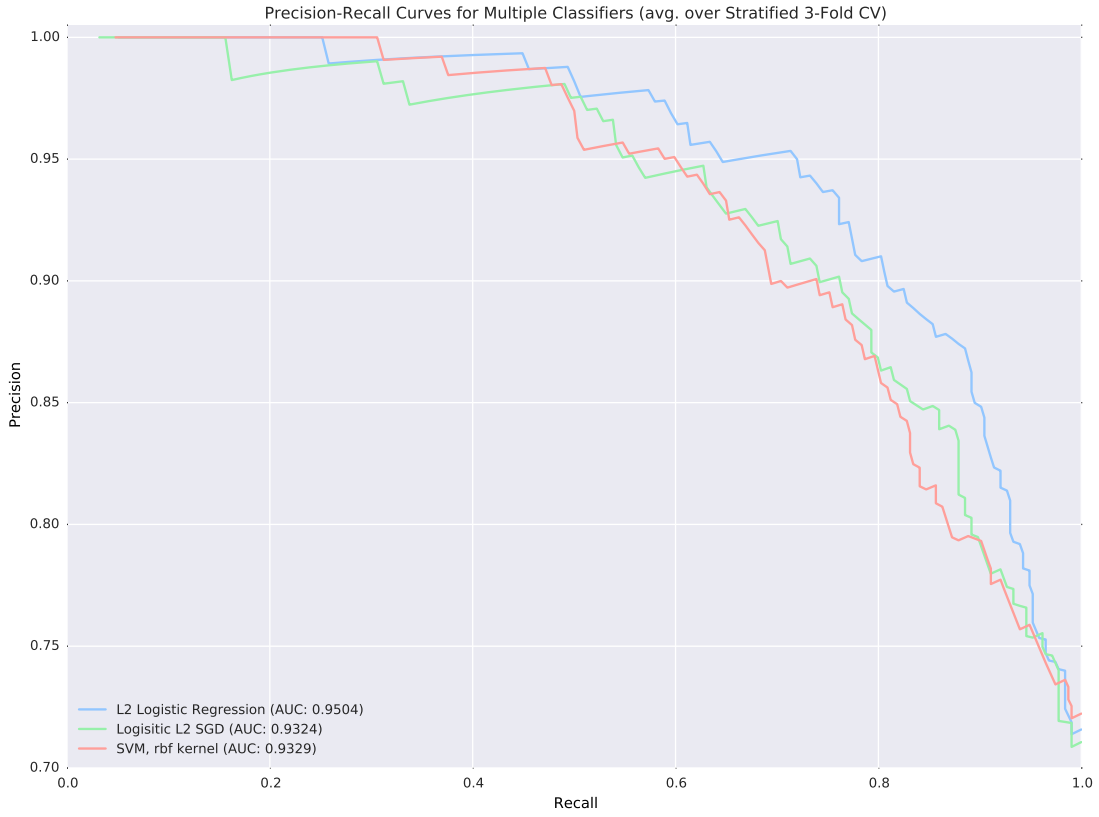
Basic performance metrics averaged over a stratified three-fold cross validation are in Figure 7.

Figure 7: Classifier comparison



While nearly all tested classifiers perform well in terms of precision, logistic regression performs significantly better for positive-class (Yes-vote classification) recall among the models that are relatively more accurate, and has the best overall accuracy. This performance is confirmed when looking at the precision-recall curve of three classifiers in Figure 8:

Figure 8: Precision-recall curves

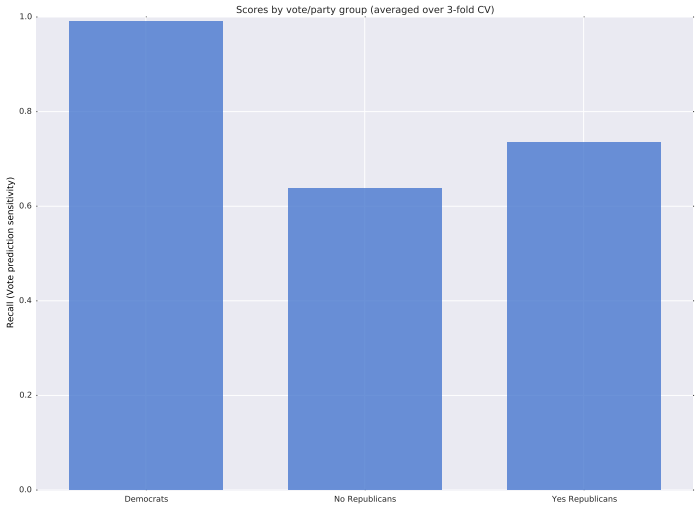


Note here that the PRAUC of the logistic regression classifier indicates its superiority over stochastic gradient descent and the support vector machine classifier, which we suspect is misrepresenting the data by way of an over-fit, high-dimensional separation boundary.

We focus on recall especially in the context of vote prediction, as in our case it is more important to most accurately represent the population across voting categories as a whole than it is to predict correctly for a specific individual. Recall of Yes-voting Republicans is

especially of interest, as they are in this case the defectors and intuitively may be harder to predict. In fact, as we observe in Figure 9, the logistic regression classifier is able to recall a greater rate of Yes-voting Republicans (and Democrats nearly perfectly), with the penalty of not retrieving as many No-voting Republicans. This imbalance in the model is in part a result of class imbalance, with No-voters making up only 30.07% of the sample space. Still, the application of a logistic regression classifier indicates that vote prediction on a by-user basis using only that users' tweet data is feasible and accurate for many tasks.

Figure 9: Average score by group



4.6 Classification over time

We investigated the temporal patterns of each group's text distances using metrics such as cosine-similarity of BOW vectors and JS distance of topic distributions. The results were unsatisfying and indeed our logistic regression approach to separating the data has proven much more able to discriminate between the yes-voters and the no-voters in the corpus. However, the classification approach presented above does not answer the temporal variation question: is there anything about the texts that evolves over time, to indicate which Republicans would defect from the party line vote?

In order to answer this question, we split the data into two parts: one corresponding to all tweets sent in the first 14 days in the period, and one to tweets after that day. While this method will not have the by-day granularity of the daily vector-distance approach, it may still reveal a shift in twitter behavior over time, if we found different separations of the voting groups in the different time periods.

We use three text corpora (entire time period, first half and second half) and repeat the logistic regression classification exercise for each. Again, we compile one document per legislator from all the tweets posted in that period. Each document is represented as a BOW vector, and the vote of that legislator at the end of the period is coded as a binary vector: $V_i = \{0\,if\,voted\,no, 1\,if\,voted\,yes\}$. Table 4 reports confusion matrices over time and over group. Contrary to our expectations, as we get closer to the end of the shutdown, we are not able to predict votes by group better. However, we are able to predict the target group, the Yes-Republicans, better using their texts from the second half.

Table 4: Confusion matrices: over time logistic regression classification

		Predicted								
		Whole month			First half			Second half		
Actual	All		No	Yes		No	Yes		No	Yes
		No	86	49	No	70	59	No	52	74
		Yes	29	285	Yes	33	271	Yes	22	270
	Dem		No	Yes		No	Yes		No	Yes
		No	0	0	No	0	0	No	0	0
		Yes	2	209	Yes	0	204	Yes	1	199
	Y-Rep		No	Yes		No	Yes		No	Yes
		No	0	0	No	0	0	No	0	0
		Yes	27	74	Yes	33	65	Yes	21	70
	N-Rep		No	Yes		No	Yes		No	Yes
		No	86	49	No	0	0	No	52	74
		Yes	0	0	Yes	0	204	Yes	0	0

We also report overall recall and precision, as well as recall and precision for each voting group separately, when using each of the three-split data.

Figure 10: Overall classification metrics over time

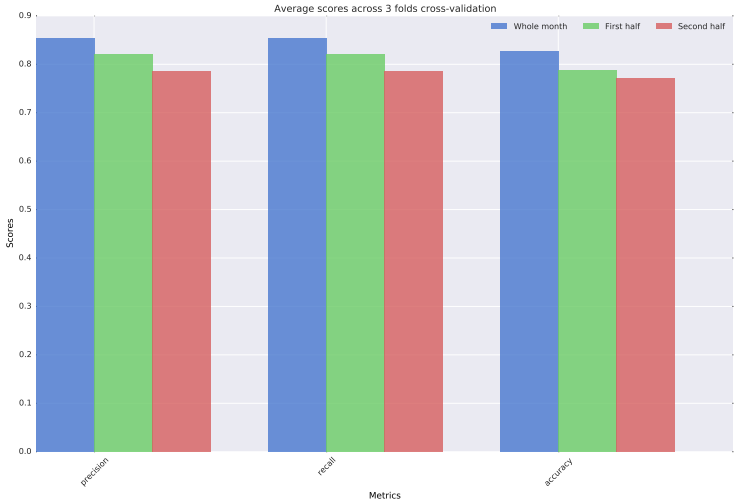


Figure 10 shows the overall classification metrics when data is used from each of the three time periods. We see that all metrics are best when using the full range data, and worst with only the second period's data. While overall metrics go down as apparent in the first sub-plot, we actually observe an increase of recall for Yes-Republicans in the second period, when compared with the first period.

Figure 11: Predicted votes over time

(a) Whole month

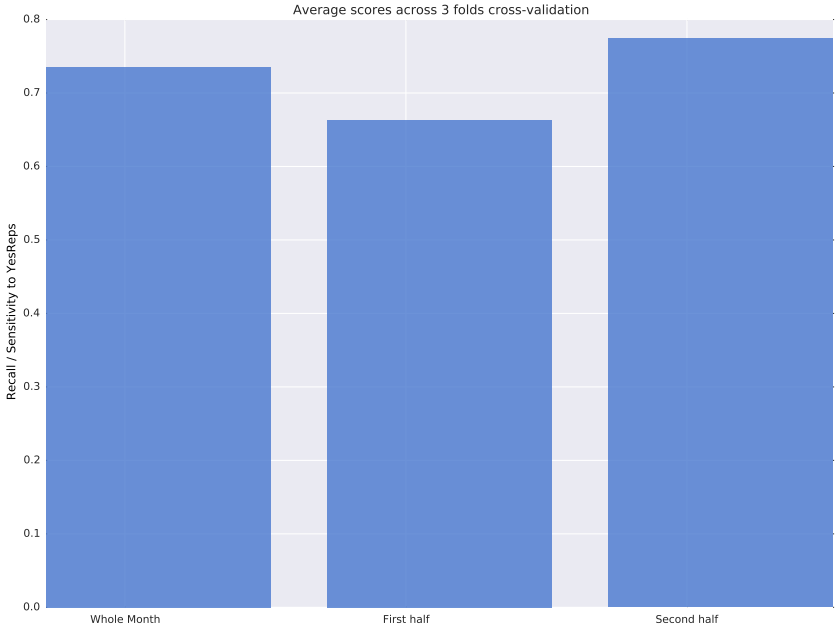
(b) First half

(c) Second half

Figure 11 shows the predicted votes for each legislator (when they were in the hold-out set) over a 3 fold cross-validation scheme. The axes are DW-Nominate, such that a distinction between the democrats (on the left) and republicans (on the right) is obvious.

We see that we are indeed able to predict the target group (the Yes-Republicans) markedly better using only their texts from the second half month. (Figure 12)

Figure 12: Yes-Republicans recall by period



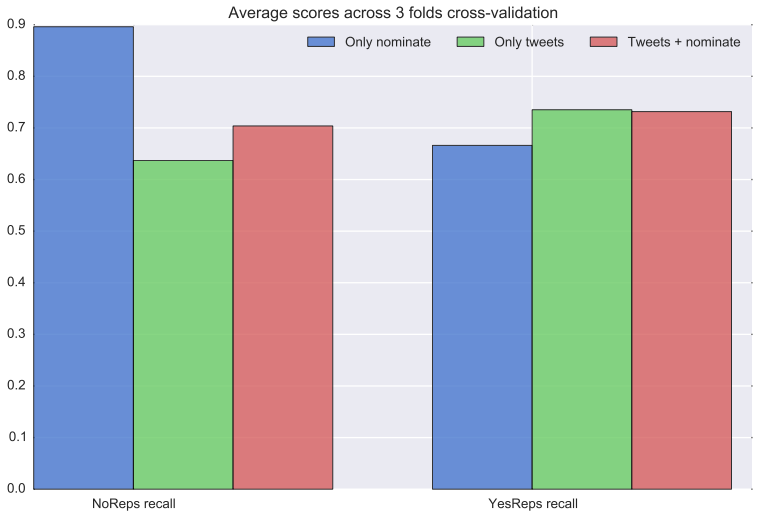
This may indicate that they have changed their behavior on social media at some point during the month preceding the vote, in a way that indicates their intention to vote with the democrats. On the other hand, the result that overall classification accuracy is lower for other voting groups seems puzzling; we see from the confusion matrices above (Table 4) that this is largely driven by a drop in no-voting republican prediction accuracy, which may suggest that all republicans alike have shifted their language use at some point during the month, which has made it harder to separate between them, when only using data from the second period. This is an issue which requires further investigation.

4.7 Supervised classification based on text and covariates

We trained a logistic regression classifier using the same parameters as before, when using only text features. Feature vectors, representing each legislator, were augmented with nominate-1 and nominate-2, and evaluation was repeated. Finally, we train and cross-validate classifiers based on DW-Nominate 1 and DW-Nominate 2 as the only features.

Results reported are averaged over three-fold cross-validation.

Figure 13: Recall tweets vs. DW-Nominate



Looking at figure 13 we see that when using only DW-Nominate as features, recall for Yes-Republicans is very high, and recall for No-Republicans is quite low. This can be explained easily by looking at a figure where votes are plotted on a 2D DW-Nominate score plane. The clear separation in the DW-Nominate space is that between republicans and democrats, and there is no clear separation between Yes-Republicans and No-Republicans. Our classifier thus learns to put all Democrats, and a few Republicans which are (in nominate terms) the most liberal, in the yes voter group. This classifies most republicans as no-voters, which puts No-Republican recall off the charts, while performing poorly at picking Yes-Republicans.

Comparing results of the DW-Nominate-only features to the results obtained using tweets as data, we come to the point we wish to make here: it appears that more of the separation between Yes-Republicans and No-Republicans is in text-space, while Yes-Republicans and No-Republicans are mixed in ideology space. DW-Nominate scores are based on voting history and indeed provide some information about which Republicans were the most likely to vote yes on this issue, but their tweets provide a clearer separation. Finally, when combining those two sources of data, we are able to achieve the best results for Yes-Republicans.

4.8 Unsupervised learning: clustering

We have seen so far that we are able to predict vote choice through supervised classification, relying on the fact that we already know the outcome of the shutdown vote. However, we would like to be able to predict votes before they happen, based on the social media posts of legislators. With this purpose in mind we move towards unsupervised classification, which is appropriate in the absence of pre-labeled data and human coding of documents into classes. The most commonly used method of unsupervised classification is clustering. The goal is to create clusters of documents such that we are maximizing similarity within each cluster and minimizing it between clusters. (Manning et al. 2009) We chose K-means clustering for its ease of interpretation, simplicity and speed. Its objective is to minimize the average squared euclidean distance of the documents in each cluster from their mean, which is the center of the cluster. Ideally, clusters should not overlap. This is equivalent to minimizing the residual sum of squares of the observations from the cluster centroids. The algorithm starts by selecting initial cluster centers randomly. It then moves them around in space in search of the minimal residual sum of squares.

Here we use Mini-batch K-means, a variant of K-Means that uses mini-batches to reduce computation time, while still attempting to optimize the same objective function. In each iteration, the algorithm takes small batches of samples and assigns each data point in the batch to a cluster. It then updates the locations of cluster centroids based on the new points from the batch. Mini-batch K-means uses a gradient descent update, which is significantly faster than a normal Batch K-Means update. We use the Python package scikit-learn to perform the clustering.

The previous analyses have shown that whereas Yes-Republicans seem to be closer to Democrats in their use of language, they differ from them in significant ways. Indeed, upon fitting models with two clusters and models with three clusters we conclude that the fit is better for those with three clusters. Several measures can be used to evaluate clustering performance. In Table 5 we report some of these measures for two K-means algorithms

with two centroids and one with three centroids. The adjusted Rand index is a function which measures the similarity of observed and predicted assignments, ignoring permutations and with chance normalization. When two partitions agree perfectly the Rand index is 1. Homogeneity score measures the extent to which each clusters contain only members with the same label. Completeness score measures the extent to which all members with the same label are assigned to the same cluster. The v-measure is the harmonic mean between homogeneity and completeness. Mutual information measures the agreement of the two assignments, ignoring permutations. The higher the values of measures are, the better the performance of the clustering algorithm.

The meaning of the clusters is intuitive in this case: two clusters could correspond to either the two parties or to the vote choice, whereas three clusters would correspond to the party-vote grouping on the issue of the shutdown.

Table 5: Clustering: 2 vs. 3 clusters

	2 clusters - party	2 clusters - vote	3 clusters - vote groups
Homogeneity	.25	.22	.30
Completeness	.35	.22	.54
V-measure	.29	.22	.42
Mutual information	.20	.13	.36
Adjusted Rand Index	.69	.64	.60

Overall, a clustering algorithm with three clusters that correspond to the three groups, Democrats, No-Republicans and Yes-Republicans performs better than either of the two algorithms with two clusters. However, the algorithm does not perform as well as we would like it to. Table 6 reports cluster assignment against observed group placement. Note that the labels for the clusters are not informative, since they are being randomly assigned. Ideally, we would like all observations in a group to be placed in a single cluster. While we do see clusters that have no observations from one of the three groups, we also have one cluster (labeled C1) with observations from all three of them. This is most likely the

case because all theses groups are focused on the shutdown debate, and there is a large amount of overlap in the language they use. However, we are able to distinguish between democrats and republicans very well: no democrats are assigned to C2, and no republicans are assigned to C0. There is also a difference in the probability of being assigned to C0 vs. C1 among republicans: Yes-Republicans are much more likely to be assigned to C1 than are No-Republicans.

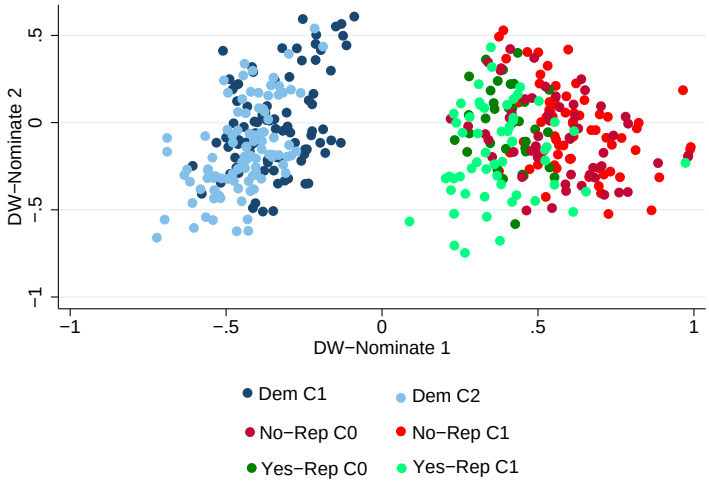
Table 6: Cluster assignment by group

	C0	C1	C2
Dem	0	88	98
No-Rep	70	55	0
Yes-Rep	35	57	0

Mini-Batch K-Means algorithm.

Figure 14 provides a visual representation of these results. We have plotted the DW-Nominate 1 score against the DW-Nominate 2 scores for each member of Congress. This way we can clearly distinguish between democrats (on the left) and republicans (on the right). Democrats are represented in shades of blue, Yes-Repubicans are represented in shades of green and No-Republicans in shades of red. Ideally, we would like to see a single shade of the same color. However, for each group we see two, which makes it relatively easy to distinguish democrats from republicans, but relatively harder to distinguish republicans among themselves. However, as the results in the table showed, we can see that relatively more Yes-Republicans are assigned to cluster C1 than No-Republicans, and more No-Republicans are assigned to cluster C0.

Figure 14: Clustering: K-means, three groups



5 Conclusion

We found that the language legislators used to tweet about the 2013 US government shutdown differed according to the voting-group they belonged to. Based only on the text of their tweets, we can distinguish between the Democrats, Yes-Republicans and No-Republicans. Our supervised classification algorithms have identified Yes-Republicans as more similar to the Democrats and have accurately classified most of them as yes-voters. Moreover, we were able to distinguish between Yes-Republicans and No-Republicans better using text than were using only traditional covariates, such as DW-Nominate scores. Combining these two types of features increased our ability to separate between the groups. An interesting characteristic of the data is the fact that groups are less distinguishable among themselves than individuals are, suggesting that aggregating the text of individual legislator obfuscates some of the features which are relevant for individual-level classification.

The results from our clustering algorithms also show that it is possible to identify the three groups based on their choice of words on social media, in the absence of any information about the actual group-membership or votes. However, having performed the analysis on a single case, for which the outcome was already known, we need to be cautious about making

inferences about our ability to make predictions. Therefore, the next step in this project will be to test the models on new data, trying to predict votes before they take place.

References

[1] Nicholas Beauchamp. Predicting and explaining supreme court decisions using the texts of briefs and oral arguments. In *APSA 2012 Annual Meeting Paper*, 2012.

[2] Nick Beauchamp. Using text to scale legislatures with uninformative voting. *Working paper*.

[3] Adam Bermingham and Alan F Smeaton. On using twitter to monitor political sentiment and predict election results. *Working paper*, 2011.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] Daniel Diermeier, Jean-François Godbout, Bei Yu, and Stefan Kaufmann. Language and ideology in congress. *British Journal of Political Science*, 42(01):31–55, 2012.

[6] Sean Gerrish and David Blei. The ideal point topic model: Predicting legislative roll calls from text. In *Proceedings of the Computational Social Science and the Wisdom of Crowds Workshop. Neural Information Processing Symposium*, 2010.

[7] Sean Gerrish and David M Blei. Predicting legislative roll calls from text. In *Proceedings of the 28th international conference on machine learning (icml-11)*, pages 489–496, 2011.

[8] Sean Gerrish and David M Blei. Predicting legislative roll calls from text. In *Proceedings of the 28th international conference on machine learning (icml-11)*, pages 489–496, 2011.

[9] Sean Gerrish and David M Blei. How they vote: Issue-adjusted models of legislative behavior. In *Advances in Neural Information Processing Systems*, pages 2753–2761, 2012.

[10] David Goldblatt and Tyler ONeil. How a bill becomes a law-predicting votes from legislation text. *Working paper*, 2012.

[11] Benjamin E Lauderdale and Tom S Clark. Scaling politically meaningful dimensions using texts and votes. *American Journal of Political Science*, 2014.

[12] Michael Laver and Kenneth Benoit. Locating tds in policy spaces: the computational text analysis of dail speeches. *Irish Political Studies*, 17(1):59–73, 2002.

[13] Brendan O'Connor, Brandon M Stewart, and Noah A Smith. Learning to extract international relations from political context. In *ACL (1)*, pages 1094–1104, 2013.

[14] Andranik Tumasjan, Timm O Sprenger, Philipp G Sandner, and Isabell M Welpe. Election forecasts with twitter: How 140 characters reflect the political landscape. *Social Science Computer Review*, page 0894439310386557, 2010.

[15] Eric Wang, Esther Salazar, David Dunson, Lawrence Carin, et al. Spatio-temporal modeling of legislation and votes. *Bayesian Analysis*, 8(1):233–268, 2013.

[16] Felix Ming Fai Wong, Chee Wei Tan, Soumya Sen, and Mung Chiang. Quantifying political leaning from tweets and retweets. In *ICWSM*, 2013.

[17] Tae Yano, Noah A Smith, and John D Wilkerson. Textual predictors of bill survival in congressional committees. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 793–802. Association for Computational Linguistics, 2012.