

An Embedding Model for Estimating Legislative Preferences from the Frequency and Sentiment of Tweets

Gregory P. Spell

Duke University / Durham, N.C.
gregory.spell@duke.edu

Brian Guay

Duke University / Durham, N.C.
brian.guay@duke.edu

D. Sunshine Hillygus

Duke University / Durham, N.C.
hillygus@duke.edu

Lawrence Carin

Duke University / Durham, N.C.
carin@duke.edu

Abstract

Legislator preferences are typically represented as measures of general ideology estimated from roll call votes on legislation, potentially masking important nuances in legislators’ political attitudes. In this paper we introduce a method of measuring more specific legislator attitudes using an alternative expression of preferences: tweeting. Specifically, we present an embedding-based model for predicting the frequency and sentiment of legislator tweets. To illustrate our method, we model legislators’ attitudes towards President Donald Trump as vector embeddings that interact with embeddings for Trump himself constructed using a neural network from the text of his daily tweets. We demonstrate the predictive performance of our model on tweets authored by members of the U.S. House and Senate related to the president from November 2016 to February 2018. We further assess the quality of our learned representations for legislators by comparing to traditional measures of legislator preferences.

1 Introduction

Legislator preferences are typically estimated as general measures of ideology using roll-call votes on legislation. However, such measures fail to capture aspects of preferences not reflected in legislation, such as attitudes towards a sitting president. For instance, Sen. Bob Corker (R-TN) famously referred to the Trump White House as an “adult day-care center,” John McCain (R-AZ) said Trump “is often poorly informed,” and Jeff Flake (R-AZ) called him a “danger to a democracy,” yet all of these Republican Senators cast more than 80% of their legislative votes in line with president (Silver and Bycoffe, 2019). Generally, the political science research recognizes that the public’s views of the president have spillover effects on evaluations of

legislators, which incentivizes strategic communication about the president. For example, Senate Majority Leader Mitch McConnell recently encouraged Republican senators in vulnerable re-election campaigns to distance themselves from Trump. Understanding legislators’ attitudes toward the president enables greater understanding and measurement of such strategic communications. Furthermore, these attitudes also matter for understanding the president’s ability to pass his legislative agenda.

In this paper, we propose a new method for estimating legislator preferences from the frequency and sentiment of their tweets using a novel combination of spatial models based on item response theory and the modeling of count data. We use this method to estimate legislator preferences about Donald Trump using tweets by members of Congress and Donald Trump in the 15-month period following election day in November 2016. In our model, legislator embeddings interact with embedding representations of Donald Trump himself, constructed from a neural network using the text (and timing) of his tweets during the same time frame. Thus, our model leverages the text feature extraction capabilities of neural networks and incorporates the legislator sentiment in tweets about Trump as well as the strategic decision about whether and when to tweet about him. We quantitatively assess the quality of our learned legislator representations by demonstrating the model’s predictive performance on a test set of tweets, and we also compare our model-obtained embeddings to DW-NOMINATE scores, traditional measures of legislator ideology.¹ Our analysis not only validates the modeling approach but also highlights that attitudes towards Trump are not being entirely captured by legislative voting behavior. More

¹In Appendix C we also include comparisons with the percent of the time legislators vote with Trump and Trump’s vote share in legislators’ districts in the 2016 election.

broadly, a method for estimating domain-specific preferences, rather than general ideological ideal points, broadens the range of hypotheses than can be tested by political researchers.

2 Measuring Legislator Preferences

The predominant method of measuring legislator preferences over the past half-century has been the modeling of the *ideal point* of a legislator from recorded votes on policy legislation. These ideal points constitute a spatial model for legislative behavior, as both legislators and policies are represented in a low-dimensional Euclidean space (Poole and Rosenthal, 1997; Clinton et al., 2004). Such ideal points are interpreted as measures of ideological preferences and have been used to test hypotheses on topics such as political polarization, political representation, and cross-institutional relationships (Tausanovitch and Warshaw, 2018).

A key limitation of initial methods for estimating ideal points was the inability to perform out-of-sample predictions. Thus, they could not be used to predict votes on new legislation. To address this shortcoming, Gerrish and Blei (2011) extended the ideal point model by placing legislation into a “political space” based upon the latent topics of the legislation’s text and perform prediction using these topics. Xing et al. (2017) use a nonparametric Bayesian model to incorporate constituency data into a factor model for legislative roll calls and text, with the text again being analyzed using a topic model. Further efforts to incorporate bill text using topic models come from Wang et al. (2010), Gerrish and Blei (2012), Nguyen et al. (2015), and Gu et al. (2014). The incorporation of text into ideal point modeling is not limited to legislators: Sim et al. (2016) model U.S. Supreme Court behavior using a generative model for amicus briefs.

Efforts to incorporate text into vote prediction were improved by moving to an embedding paradigm rather than topic models. Kraft et al. (2016) incorporate word embeddings into a model for vote prediction by representing a piece of legislation as the average of its word embeddings and further representing legislators using *ideal vectors* as a multi-dimensional extension to ideal points. Kornilova et al. (2018) augment bill text with bill metadata (i.e., bill sponsor information) to improve the predictive capabilities of legislator embeddings, and use a convolutional neural network (CNN, Kim (2014)), rather than the average over bill word em-

beddings, to model bill text.

While tweets have increasingly been used to measure political preferences of the mass public (Wang et al., 2016; Preoțiuc-Pietro et al., 2017), little attention has been paid to the potential of using tweets to measure legislators’ preferences. One notable exception, Barbera (2015), uses the structure of social networks on Twitter to learn ideological positions of both political elites and the general public, but does not incorporate information from the tweets themselves. As all legislators in the U.S. House and Senate now use Twitter to communicate with constituents on a wide variety of topics, we recognize an opportunity to observe nuances in legislator preferences not captured by broader ideological measures that rely on roll call votes.

Here we focus specifically on legislators’ attitudes toward the sitting president. While attitudes toward the president are among the most frequently measured aspects of public opinion, there is currently no method for explicitly measuring these preferences among legislators. We develop an embedding model that jointly predicts the frequency and sentiment of legislator tweets about Donald Trump. Similar to the Kraft et al. (2016) modeling of legislator votes in response to the text of legislation, here legislator tweets are considered as a response to text features extracted from Donald Trump’s tweets. Whereas embedding models for vote prediction analyze only one outcome of legislator behavior (i.e., the vote itself), our embedding model is trained to predict multiple outcomes of legislator behavior in both tweet counts and content in the form of sentiment. Moreover, because our model does not rely on votes cast by legislators, it could be used to estimate preferences among a wider range of political actors (e.g., candidates, cabinet members) on a variety of domains, and with texts other than tweets.

3 Tweet Dataset

We obtained all publicly-available tweets by members of Congress from TweetCongress, a Sunlight Foundation initiative. We restricted the sample to only those tweets that contained any of a specific set of terms related to Donald Trump (in addition to his Twitter handle): “Donald Trump,” “Trump,” “realDonaldTrump,” “MAGA,” (an acronym for Trump’s campaign slogan “Make America Great Again”) “whitehouse,” “WhiteHouse,” “POTUS,” (acronym for “President of the United States”), and

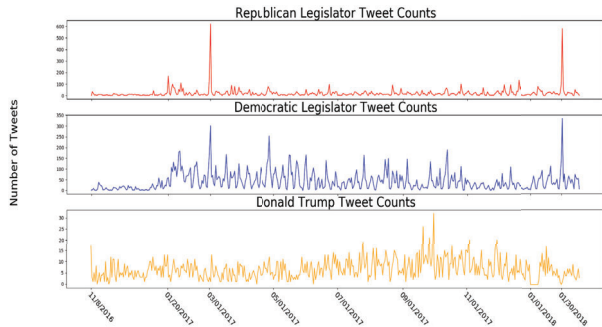


Figure 1: Number of tweets by Republican and Democratic legislators about Trump, as well as tweets by Trump, over time.

“potus.” Of these, we further restricted the tweets to span in time from November 2016 to February 2018, when the data was collected. This culling process yielded 29,696 tweets from 451 legislators.

The model also incorporates tweets from Trump, which we obtained from the website www.trumptwitterarchive.com. For each day included in the dataset, the text of all tweets by Donald Trump was agglomerated and preprocessed by removing excess whitespace and lowercasing all letters. The text was tokenized and each word-token mapped to an integer identifier, with a vocabulary mapping of 2783 words. For each day, we obtain a sequence of integers representing the words composing the text of Donald Trump’s tweets from that day, and these are the inputs to the model described in Section 4.3.

Figure 1 plots the number of tweets by Republicans, Democrats, and Trump over time for the period we examine. There were only 13 days for which Trump did not tweet (2.79%), and the most tweets that he sent in a single day was 32. The most tweets by a Democrat in a single day was 99, while the most tweets by a Republican in a single day was 25. The variation in tweets across time highlights one of the key features of the model—the incorporation of not only the sentiment of tweets about Trump but also the number of daily tweets.

Of the 29,696 Trump-related legislator tweets, a subset of 4,661 tweets were randomly selected to be manually labeled with respect to their sentiment about Trump, using a three-point “positive,” “negative,” “neutral” scale based on the text of the

tweet² from November 2016 to February 2018.

We divided the tweets temporally by day into disjoint training, validation, and test sets, such that all tweets from each day were randomly assigned to one of the three sets. The training, validation, and test sets contain 70%, 10%, and 20% of all days, respectively. Table 1 outlines how many days and tweets are included in each set.

	# Days	# Labeled	# Total
Training	324	3069	20116
Validation	47	412	2441
Test	95	1180	7139

Table 1: Split of Training, Validation, and Test sets.

4 Legislator Tweet Model Formulation

Our proposed model combines an embedding model for legislators with models for ordinal and count data, predicting both the number of daily tweets about Donald Trump sent by each legislator and the sentiment of labeled tweets. The joint nature of this model not only enables a more nuanced representation of legislators, but also accounts for the fact that a legislator who consistently tweets in favor (or against) the president is different from one who tweets occasionally, even if both express similar sentiment. The model also incorporates text features from Trump’s tweets to provide context for legislator tweets as reactions to Trump.

Underpinning our model is the assumption of a latent political space of dimension K . In this space, we learn a set of “day embeddings” (or “Trump embeddings”) that interact with a set of legislator embeddings. For a particular day t , let $\tau_t \in \mathbb{R}^K$ be a vector that represents Donald Trump on that day. Indexing legislators by $i \in \{1, 2, \dots, N\}$, we endow a legislator i with a vector $\mathbf{v}_i \in \mathbb{R}^K$ as well as a bias term $b_i \in \mathbb{R}$, which captures a legislator’s propensity to react to Trump regardless of how he presents himself via Twitter. While the legislator embeddings are learned as free parameters of the model, the Trump embeddings are constructed using the text of Donald Trump’s tweets. We now describe how we use legislator and Trump embeddings to predict tweet counts and sentiment.

²See Appendix A for full description of the dataset construction.

4.1 Tweet Count Model

Understanding legislator-president interactions requires understanding not only the sentiment of legislators’ remarks about the president, but also whether and how often they remark about him. This distinguishes, for instance, a legislator who criticizes Trump every week from one who criticizes Trump only once during his tenure. Even when the sentiment expressed in these two legislators’ tweets is identical, the fact that one legislator expresses that sentiment more frequently likely reflects a more negative attitude toward Trump. Furthermore, since we model tweets as a response to a daily representation of Trump, modeling counts reveals legislators who respond in concert with each other and may share similar preferences.

Let x_{it} be the number of tweets that legislator i sends about Donald Trump on day t . We consider two distributions with which to construct our tweet count model: Poisson and Negative Binomial. While the former offers simplicity, the latter is more flexible and suitable for overdispersed data because of its additional parameter. In Section 5, we compare the Poisson and Negative Binomial model performances. We will parameterize the Negative Binomial using (p_{it}, r) :

$$x_{it} \sim \text{NegBin}(p_{it}, r), \quad p_{it} = \sigma(\boldsymbol{\tau}_t^\top \mathbf{v}_i) \quad (1)$$

where $\sigma(\cdot)$ is the sigmoid function defined by $\sigma(x) = \frac{1}{1+\exp(-x)}$, which is used to transform the input onto $(0, 1)$ to represent a probability. The remaining parameter r is learned as a common free parameter for all legislators and days. For the Poisson case, we model the rate parameter of the distribution as the exponential of the dot-product between the Trump and legislator embeddings. This choice ensures the rate parameter is non-negative while also modeling an “interaction” between legislators and Trump.

We train the count model by minimizing the negative log-likelihood (NLL) of the training data under either of the assumed distributions. We denote the total count-loss over a training set \mathcal{X}_{tr} as:

$$\mathcal{L}_{count} = \sum_{x_{it} \in \mathcal{X}_{tr}} \text{NLL}_{count}(x_{it}; \boldsymbol{\tau}_t, \mathbf{v}_i) \quad (2)$$

4.2 Tweet Sentiment Model

Let y_{it} be an ordinal variable that encodes the sentiment legislator i expresses in a tweet about Donald

Trump on day t . We consider an ordinal model to account for the possible gradations of approval. Assuming L sentiment levels, the model is parameterized by a set of cutpoints, $\mathcal{C} = \{c_0 < c_1 \leq c_2 \leq \dots \leq c_{L-1} < c_L\}$, where c_0 and c_L are defined to be $-\infty$ and ∞ , respectively. The remaining cutpoints are learned during model training.

Let $z_{it} \in \mathbb{R}$ be a latent variable underlying the ordinal response. Then for a thresholded ordinal model, the predicted sentiment takes value l for which: $c_{l-1} < z_{it} < c_l$. Under a cumulative link model (CLM)³ for ordinal regression, the predicted probability of a particular sentiment level l is:

$$p(y_{it} = l | z_{it}; \mathcal{C}) = \sigma(c_l - z_{it}) - \sigma(c_{l-1} - z_{it}) \quad (3)$$

where again $\sigma(\cdot)$ is the sigmoid function. The latent variable z_{it} is a function of the attributes of legislator i and of Trump at day t . As with the count model, we seek to employ a map that captures the interaction between the legislator and Trump embeddings, and thus we employ a weighted inner product. Additionally, we expect that legislators maintain a concrete bias towards Trump, which we include in the term b_i for each legislator. Thus, we obtain the variable z_{it} through the following map:

$$z_{it} = g(\mathbf{v}_i, \boldsymbol{\tau}_t, b_i) = \boldsymbol{\tau}_t^\top \mathbf{H}_g \mathbf{v}_i + b_i \quad (4)$$

where $\mathbf{H}_g \in \mathbb{R}^{K \times K}$ is a learned weight matrix.

As with the count model, the sentiment model is trained by optimizing the negative log-likelihood of the sentiment-labeled tweets in the training set. With the predicted probability of the correct label, $p(y_{it} = l)$, given by equation 3, and the set of all labeled tweets in the training set being \mathcal{Y}_{tr} , then the total loss for the sentiment model is given by:

$$\mathcal{L}_{sent} = \sum_{y_{it} \in \mathcal{Y}_{tr}} \sum_{l \in \{1, 2, \dots, L\}} -\mathbb{I}(y_{it} = l) \log p(y_{it} = l) \quad (5)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, in which $\mathbb{I}(\cdot) = 1$ when the argument is true and 0 otherwise.

4.3 Trump Embedding Construction

In the ideal point/vector models that consider roll call data, legislator behavior is a response to policies as captured by the text of bills. As we seek an alternative to legislation as a method of measuring

³See Gutiérrez et al. (2016) for ordinal model overview.

preferences, we rely instead on Twitter behavior but similarly construct embeddings that legislators respond to. Since Donald Trump is our entity of investigation, we use the text of his tweets to construct such embeddings.

To map Donald Trump’s tweet text to a political embedding representation, we employ a Simple Word-Embedding Model (SWEM), (Shen et al., 2018). SWEMs rely upon word embeddings (Bengio et al., 2003; Mikolov et al., 2013) and pooling operations to encode the compositionality of text without the heavy parameterization required of such models as recurrent neural networks (RNNs, see Socher et al., 2011) or CNNs (Kalchbrenner et al., 2014; Kim, 2014). Endowing each word-token u_i in a lexicon with an embedding $\mathbf{w}_i \in \mathbb{R}^d$, we may represent a sequence of n words as a matrix of stacked embeddings: $\{\mathbf{w}_1, \dots, \mathbf{w}_L\} = \mathbf{W} \in \mathbb{R}^{n \times d}$. To extract the most salient features from every word-embedding dimension, we employ a max-pooling operation, which amounts to a column-wise maximum of matrix \mathbf{W} . Supposing that \mathbf{W}_t contains the embeddings from *all* Donald Trump tweets on day t , then we will denote $\boldsymbol{\alpha}_t \in \mathbb{R}^d$ as the max-pooled vector, These text features are subsequently mapped to the daily Trump vector by an affine transformation:

$$\boldsymbol{\tau}_t = \mathbf{M}\boldsymbol{\alpha}_t + \mathbf{a} \quad (6)$$

where $\mathbf{M} \in \mathbb{R}^{d \times K}$ and $\mathbf{a} \in \mathbb{R}^K$ are a weight matrix and bias vector that are shared by all days t . This transformation can be made more flexible by introducing a non-linear activation function, $\phi(\cdot)$, such as the rectified linear unit (ReLU). This non-linear “hidden” layer is described by:

$$\boldsymbol{\tau}_t = \mathbf{M}_2 \phi(\mathbf{M}_1 \boldsymbol{\alpha}_t + \mathbf{a}_1) + \mathbf{a}_2 \quad (7)$$

where an additional weight matrix and bias vector have been appended.

4.4 Model Training & Parameter Learning

The parameters in the model to be learned include the legislator embeddings and biases, the word embeddings, the parameters of the maps to count and ordinal variables, and the parameters of the map from text features to Trump embeddings. We refer to this collection as Θ . The optimization objective is the combination loss of the negative-log likelihood of the count and ordinal models:

$$\mathcal{L}(\Theta) = \gamma \mathcal{L}_{count} + (1 - \gamma) \mathcal{L}_{ord} \quad (8)$$

where \mathcal{L}_{count} and \mathcal{L}_{ord} are given by equations 2 and 5, respectively, and γ is a hyperparameter that controls the relative importance of the two component losses. The construction of equation 8 allows the researcher to only admit tweet count information by setting $\gamma = 1$ and only admit tweet sentiment information by setting $\gamma = 0$; a balance may be achieved by choosing $\gamma \in (0, 1)$. The Adam algorithm (Kingma and Ba, 2015) is used for gradient-based optimization of 8 with a learning rate of $\eta = 10^{-4}$.

5 Predictive Results

To demonstrate the efficacy of our model for legislator tweeting behavior with respect to President Donald Trump, we first show that the construction of Trump embeddings from the language of his own tweets provides an informational signal for legislators to react to. We train our model using the days for the training set and present the predictive results for days in the test set. Since the model seeks to capture two aspects of legislator tweeting behavior, we evaluate the model using two metrics: the negative-log likelihood of the count model and the mean-absolute-error (MAE) of the sentiment model. Overall model performance is also captured by the total loss of the model, which is the weighted negative-log likelihood of both the count and sentiment models, equation 8. MAE is used rather than accuracy to account for the ordinal nature of the sentiment model.

The hyperparameter γ controls the balance between the two components of our model, counts and sentiment. We present our results for three settings of γ , which allows us to analyze the two components of our model separately before analyzing the joint model. A full description of the process used to tune hyperparameters and a comparison of the model with linear and nonlinear text maps can be found in Appendix B. For all results presented here, we set $K = 2$, and use a linear text map. The number of epochs for which the model was trained varies depending on model setting, but in all cases each training batch comprises 128 tweets. The model was implemented in TensorFlow (Abadi et al., 2015) and trained on a single NVIDIA Titan X GPU. Code can be found on the author’s Github at: github.com/gspell/CongressionalTweets.

5.1 $\gamma = 1$ (only count model):

When $\gamma = 1$, only the loss from the part of the model that handles tweet counts contributes to the total loss in equation 8. We present the final negative log-likelihood of the count model for both the Poisson and Negative Binomial models described in Section 4.1, and for both the case in which the text of Donald Trump’s tweets is used to construct his daily embedding representation and the case in which the Trump embeddings are free parameters of the model. For the negative binomial model, the model was trained for 75 epochs, which was the amount of training required to perform best on the validation (rather than test) set of tweets. The Poisson model was trained for 100 epochs while using the text and 2000 epochs without text. The predictive results are shown in Table 2.

	Text		No Text	
	Loss	MAE	Loss	MAE
Poisson	20,882	0.692	49,128	0.693
Neg. Bin.	16,692	0.726	18,461	0.696

Table 2: Predictive evaluation metrics on test for our model with $\gamma = 1$. Note that because only the count loss is being optimized, MAE does not reflect model performance here. Best model result bolded.

Modeling legislator tweet counts using the Negative Binomial distribution achieves superior performance to modeling using the Poisson distribution, as the Negative Binomial can better accommodate the overdispersion in the tweet counts. Additionally, using the text of Donald Trump’s tweets to construct his daily embedding that legislator embeddings interact with provides significantly better results than neglecting the text and allowing the Trump embeddings to be free parameters of the model. This effect is more pronounced for the Poisson distribution, but is present for the Negative Binomial model as well. Indeed, this aspect of the model is to be expected, since the model is evaluated on days of which there are no examples in the training set. Without using the text, there is no way for the model to represent an “unseen” day.

5.2 $\gamma = 0$ (only sentiment model):

When $\gamma = 0$, only the loss from the part of the model that handles legislator tweet sentiment contributes to the total loss in equation 8. We present the final model loss — which is the negative log-likelihood of the sentiment model — as well as the model MAE. Again, we show results for the case

in which Trump’s tweet text is used to construct embeddings and the case in which the text is not used. We also toggle an additional model setting for analysis: the inclusion of the legislator bias term, b_i , from equation 4. We adjust the number of epochs to 150 for training with text. We train the model without text for 1000 and 3000 epochs, including and excluding the legislator bias term, respectively. The results are presented in Table 3.

	Text		No Text	
	Loss	MAE	Loss	MAE
No Bias	549.63	0.140	1714.48	0.878
Bias	548.80	0.140	831.24	0.390

Table 3: Predictive evaluation metrics on test for our model with $\gamma = 0$. Best model result with respect to MAE is bolded. Comparison between the sentiment model with/without the legislator bias term as well as with/without Trump tweet text

In addition to the MAE of the ordinal model, we note that the model accuracy — which is more intuitive but less exact than MAE — is 88.4% for the best performing model, when both the legislator bias and Trump’s tweet text are used. Note that when the text of Donald Trump’s tweets is used, the model performs as well with respect to MAE with the inclusion of the legislator bias as without it. Additionally, when the bias term is included but Trump’s text is excluded, the model is able to achieve better performance than when both the text and bias term are excluded. In fact, for the case of no Trump text and no legislator bias, the model is incapable of achieving test MAE better than how it performs upon initialization. We note that while the model does train, performance on the test (and validation) never improves in that case.

Table 3 suggests that the legislator bias (when present) accounts for much of the model’s ability to predict legislator tweet sentiment, since the model achieves decent results even when no Trump text is used to construct meaningful Trump embeddings to interact with the trained legislator embeddings. Without the bias term, the interaction between Trump and legislator embeddings is the only means toward predicting tweet sentiment, which is why the necessity of text is so critical in that case.

5.3 $\gamma = 0.03$ (both counts & sentiment):

For any other value of $\gamma \in (0, 1)$, the total loss in equation 8 will have contributions from both the count and sentiment losses, and thus both aspects of the model are trained jointly. Using the validation

set, we determined that setting $\gamma = 0.03$ achieves a good balance between both the count and sentiment parts of the model⁴, obtaining a good MAE without neglecting modeling of the counts. Given the considerations discussed for $\gamma = 0, 1$, we only examine the Negative Binomial count model and the inclusion of the legislator bias term. When the model was trained using the text of Donald Trump’s tweets, it was trained for 200 epochs, while it was trained for 1500 epochs when the text was not used, and the runtimes were 3.12 and 12.9 minutes, respectively. The joint model performance is shown in Table 4, with MAE, total loss, and unweighted count model negative log-likelihood shown.

	Count NLL	MAE	Total Loss
No Text	28,571	0.213	1583.97
Text	16,782	0.127	994.76

Table 4: Predictive evaluation metrics on test for our model with $\gamma = 0.03$. Best model result bolded

As with the cases for $\gamma = 0, 1$, we have found that for our final model configuration with $\gamma = 0.03$, model predictive performance is superior when Donald Trump’s tweet text is used to construct his daily embedding representation. Additionally, the MAE on the test set for $\gamma = 0.03$ is less than the MAE for the case that $\gamma = 0$ when only the sentiment model is trained. This demonstrates that the inclusion of tweet count information mitigates sentiment prediction as well, since more information is being used to model legislators.

6 Legislator Embeddings

Training our legislator tweeting model yields a key byproduct: the legislator embeddings. As with previous spatial representations of legislator preferences, our model enables the visualization of the positions of legislators in space. In Figure 2 we plot the two dimensions of legislator embeddings from the model presented in Table 4.⁵

The most noticeable characteristic of the embeddings is how they separate legislators across party lines into Democrats and Republicans, even though party affiliations were not incorporated into the model. In the first dimension, senators are perfectly separated by party with the exception of five Democrats who have *lower* values on the first

⁴See Appendix B.2 for a discussion of choosing γ .

⁵See Appendix B.1 for a discussion of selecting model dimension

Legislator Embeddings

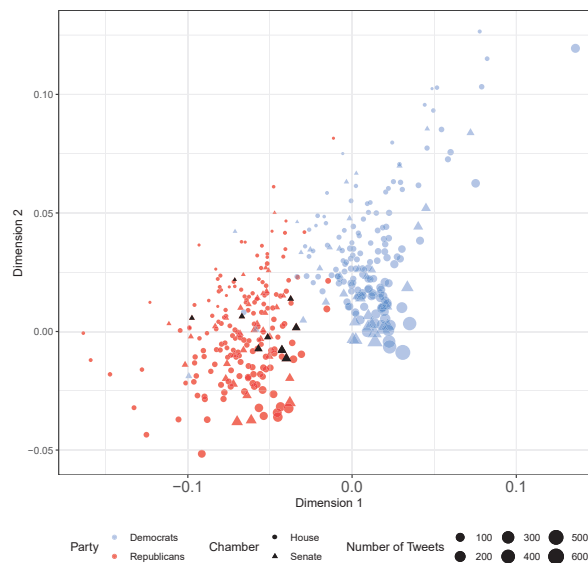


Figure 2: Learned legislator embeddings. Legislators are identified by party, chamber, and the number of tweets authored about Trump. The darker points indicate known Senate Republican critics of Trump.

embedding dimension than John McCain, the Republican Senator with the highest value: Dianne Feinstein, Heidi Heitkamp, Claire McCaskill, Angus King, and Joe Manchin. Excepting Dianne Feinstein, these senators are generally considered to be more conservative Democrats.

In the figure we also see that embeddings are not simply an artifact of the number of tweets about Trump authored by the legislator, nor whether the legislator is a member of the House or Senate. Legislators with more extreme values of Twitter sentiment relative to other members of their party can be found in both chambers of Congress and range from having authored fewer than 100 tweets about Trump to over 500.⁶

Another initial validating characteristic of the embeddings is the clustering of prominent Republican senators who have been publicly critical of Trump. We examine the spatial positions of Republican senators whom a 2017 *Washington Post* analysis identified as critical of the President based on their responses to controversial events in Trump’s presidency, such as Trump’s firing of FBI Director James Comey and response to the Charlottesville protests, as well as overall rhetoric used when discussing Trump (Lewis et al., 2017). The positions

⁶For visualization, legislators with fewer than 5 tweets are omitted from Figure 2.

Comparisons of Embeddings with: DW-NOMINATE Voting with Trump

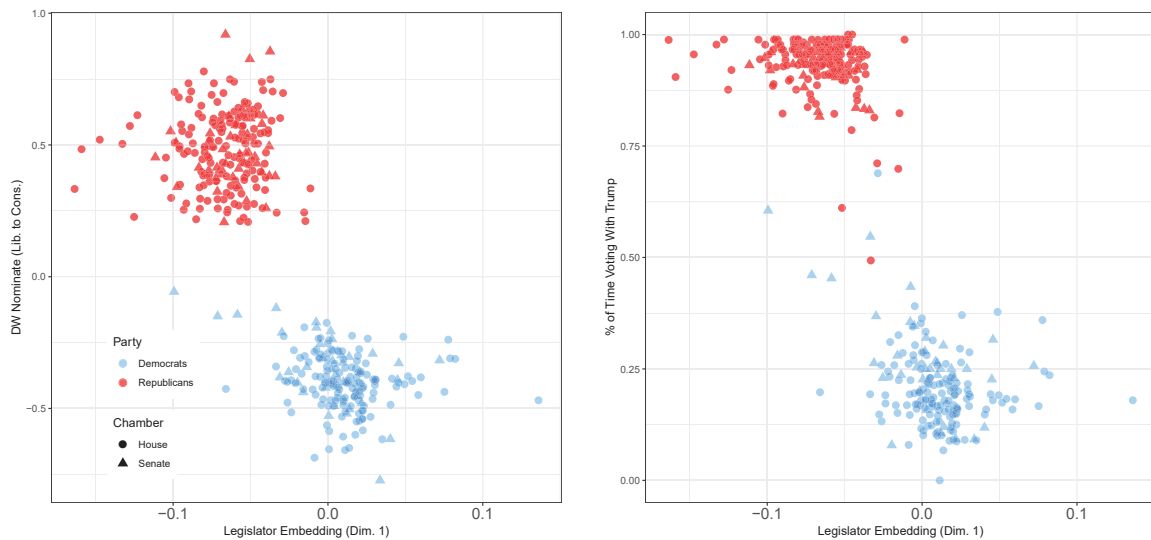


Figure 3: Comparisons of the learned legislator embeddings to DW-NOMINATE (left) and percentage of time voting with Trump (right) as measures of legislator preferences.

of these senators are highlighted (dark black points) in Figure 2. These senators are clustered together in the two-dimensional embedding space: John McCain, Jeff Flake, Joe Hoeven, Bob Corker, Marco Rubio, Shelley Capito, Dean Heller, Dan Sullivan, Lamar Alexander, and Lisa Murkowski.

Considering which Democratic legislators are interspersed near the cluster of Republicans in the two-dimensional embedding space is also informative. The two most extreme Democratic outliers were Angus King, an Independent Senator from Maine who caucuses with the Democratic party but has openly considered caucusing with the Republican party and Joe Manchin, a notably conservative Democratic senator in whose state Trump won 68.5% of the vote. We observe fewer outliers among Republicans. Among the most extreme outliers, are Ileana Ros-Lehtinen and Carlos Curbelo, whose districts Hillary Clinton won in 2016 by 19.6 and 16.3 percentage points, respectively. Ros-Lehtinen, in particular, tweeted many scathing responses to Trump regarding his controversial stance on immigration.

We next compare the embeddings to an existing measure of general legislator preferences. Figure 3 illustrates the relationship between the first dimension of DW-NOMINATE — a canonical measure of legislator ideology in political science — and the first dimension of our learned legislator embeddings. Generally, legislators who are ideologically

conservative have lower embedding values, whereas liberals have higher values. At the same time, this comparison does identify legislators who are more or less critical of Trump than might be expected based on ideology alone, thereby offering new empirical leverage to scholars examining the behavior and attitudes of legislators.

Figure 3 also compares our legislator embeddings to an approximation of how legislators might feel toward Trump: the proportion of time that they vote in line with him during the period in which legislator tweets were collected. This metric was calculated using a dataset published by Fivethirtyeight and includes only legislation on which the Trump administration publicly expressed a clear position (Silver and Bycoffe, 2019). While this measure is limited by many of the same constraints as other vote-based measures (e.g., DW-NOMINATE), it is the closest existing measure of legislators’ attitudes toward Trump. In the right panel of Figure 3, we observe little variation in the extent to which legislators vote with Trump, particularly for Republicans. Indeed, many of the President’s most prominent critics frequently voted with the president during this time period. For instance, John McCain voted with Trump 85% of the time, Bob Corker voted with Trump 84% of the time, and Jeff Flake voted with Trump 83% of the time. Meanwhile, we observe far more variation in legislator embeddings among both Republicans and Democrats. In Ap-

pendix C, we further compare our embeddings to alternative measures of legislative preferences: Campaign Finance Scores (Bonica, 2018) and Trump vote-share in a legislator’s constituency during the 2016 presidential election.

7 Conclusion

In this paper, we modeled legislator tweeting behavior towards Donald Trump, predicting the frequency and sentiment of their tweets. The proposed model yields embedding representations for legislators that we interpret as measures of legislator attitudes towards Trump. Our application suggests that ideal points estimated from roll call votes can miss this critical aspect of political preferences for members of Congress. Whereas legislative voting might recover ideological similarities and differences with the president, it is not well suited to measure attitudes toward the president orthogonal to policy preferences, such as criticisms of his rhetoric and tone. To address this shortcoming and obtain representations of legislators’ attitudes toward Trump, we have proposed a model that assigns a vector to each legislator based on the content of their tweets about Trump. We similarly represent Donald Trump with a vector for each day he tweets, constructed using the text of his daily tweets. Legislator vectors and Trump vectors interact to produce predictions of both the sentiment of legislator tweets about Donald Trump and the number of tweets produced each day. From this model we obtain representations of legislators that capture their attitudes toward the president.

Our model’s predictive performance is robust to a variety of settings and achieves sentiment predictive performance of 0.127 mean-absolute-error and 89.3% accuracy, demonstrating its capability to predict legislator tweeting behavior. When visualizing the two dimensions of learned legislator embeddings we find that the model separates legislators across party lines (despite not being trained on the party of legislators) and groups together Republican senators who are well-known critics of Trump (despite overwhelmingly voting with him on legislation).

Though our model demonstrates the capability of representing legislators’ attitudes toward Trump and performs well with respect to predicting tweet counts and sentiment based upon Donald Trump’s tweets, our method has some limitations. For one, as is the case for Rheault and Cochrane (2020), our

model is not able to produce uncertainty bounds, as deriving uncertainty measures from neural networks remains an open area of research without a clear solution within the field of machine learning.⁷ An avenue for improving the model is to allow it to capture legislators’ dynamic attitudes toward Trump over time. While legislator attitudes are currently modeled as *static* embeddings, allowing each legislator’s embedding to change over time would enable the exploration of temporal dynamics and hypothesis testing about when legislators are more likely to tweet negatively about Trump, what factors contribute to a legislator’s decision to tweet about Trump, and how the Trump’s tweets interact with legislator’s tweets over time.

While our aims in this paper were to develop a method of modeling attitudes toward Trump beyond legislative policy preferences, this method can be used to test a wide range of hypotheses about modern U.S. politics. Legislator embeddings can be used to explore how legislators appeal to different audiences, such as party leaders and constituents. The method presented here could similarly be used to evaluate how members of Congress are punished and rewarded in elections for their criticism of praise of the president. Moreover, because our model does not rely on roll call votes, it can also be used to model attitudes by any of the growing number of political elites using Twitter, such as non-incumbent political candidates, state legislators, and pundits. Possible extensions of this work could investigate enriching Trump vectors by incorporating other sources of text, such as White House press releases and speeches. While we restrict ourselves to Twitter data in this paper to maintain consistency across the sources of data for vectors representing Trump and legislators, the incorporation of auxiliary text data could provide additional context.

Acknowledgments

We thank Chris McCroskey, Tweet Congress, and Todd Shields for enabling access to the sample of legislator tweets. Gregory Spell thanks David Carlson for his invaluable mentorship and encouragement in practicing the craft of Deep Learning.

⁷To be sure, we are not the only ideal point method to share this limitation. See e.g., Imai et al. (2016).

References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org.
- Pablo Barbera. 2015. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. *Political Analysis*.
- Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*.
- Adam Bonica. 2018. Inferring roll-call scores from campaign contributions using supervised machine learning. *American Journal of Political Science*.
- Joshua Clinton, Simon Jackman, and Douglas Rivers. 2004. The statistical analysis of roll call data. *The American Political Science Review*.
- Skyler J. Cranmer and Bruce A. Desmarais. 2017. What can we learn from predictive modeling? *Political Analysis*, 25(2):145–166.
- Sean Gerrish and David M. Blei. 2011. Predicting legislative roll calls from text. *International Conference of Machine Learning* 28.
- Sean Gerrish and David M. Blei. 2012. How they vote: Issue-adjusted models of legislative behavior. *Advances in Neural Information Processing Systems* 25.
- Yupeng Gu, Yizhou Sun, Ning Jiang, Bingyu Wang, and Ting Chen. 2014. Topic-factorized ideal point estimation model for legislative voting network. *KDD 2014*.
- P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez. 2016. Ordinal regression methods: Survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146.
- Kosuke Imai, James Lo, and Jonathan Olmsted. 2016. Fast estimation of ideal points with massive data. *American Political Science Review*, 110(4):631–656.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blumson. 2014. A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Conference on Empirical Methods in Natural Language Processing*.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *Proceedings of the 3rd Annual International Conference for Learning Representations*.
- Anastassia Kornilova, Daniel Argyle, and Vlad Eidelman. 2018. Party matters: Enhancing legislative embeddings with author attributes for vote prediction. *Proceedings of the Conference for the Association for Computational Linguistics*.
- Peter Kraft, Hirsh Jain, and Alexander M. Rush. 2016. An embedding model for predicting roll-call votes. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Nicole Lewis, Amber Phillips, Kevin Schaul, and Leslie Shapiro. 2017. Where Republican senators stand on President Trump. Available online at https://www.washingtonpost.com/graphics/politics/senate-trump-support/?noredirect=on&utm_term=.711bfd810514.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*.
- Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, and Kristina Miler. 2015. Tea party in the house: A hierarchical ideal point topic model and its applications to republican legislators in the 112th congress. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.
- Keith T. Poole and Howard Rosenthal. 1997. *Congress: A Political-Economic History of Roll Call Voting*. Oxford University Press.
- Daniel Preoțiuc-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond binary labels: political ideology prediction of twitter users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 729–740.
- Ludovic Rheaault and Christopher Cochrane. 2020. Word embeddings for the analysis of ideological placement in parliamentary corpora. *Political Analysis*, 28(1):112–133.
- Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling

mechanisms. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Nate Silver and Aaron Bycoffe. 2019. Tracking Congress In The Age Of Trump. Available online at <https://projects.fivethirtyeight.com/congress-trump-score/>.

Yanchuan Sim, Bryan Routledge, and Noah A. Smith. 2016. Friends with motives: Using text to infer influence on SCOTUS. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1733, Austin, Texas. Association for Computational Linguistics.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. *Proceedings of the 28th International Conference on Machine Learning*.

Chris Tausanovitch and Christopher Warshaw. 2018. Does the Ideological Proximity Between Candidates and Voters Affect Voting in US House Elections? *Political Behavior*, 40(1):223–245.

Eric Wang, Dehong Liu, Jorge Silva, David Dunson, and Lawrence Carin. 2010. Joint analysis of time-evolving binary matrices and associated documents. *Advances in Neural Information Processing Systems* 23.

Yu Wang, Richard Niemi, and Jiebo Luo. 2016. Tactics and tallies: A study of the 2016 u.s. presidential campaign using twitter ‘likes’. *KDD*.

Zhengming Xing, Sunshine Hillygus, and Lawrence Carin. 2017. Evaluating U.S. Electoral Representation with a Joint Statistical Model of Congressional Roll-Calls, Legislative Text, and Voter Registration Data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’17, page 1205–1214.

A Data Description

In Section 3, we describe how tweets were selected for our dataset. We provide more information about the dataset here. In total, legislators sent 29,696 tweets about Trump. On average, each legislator sent 65.84 tweets about Trump, though there is substantial variation (standard deviation = 88.78, median number of tweets about Trump = 36.0).

Democratic legislators tweeted about Trump more than twice as much as Republicans. The mean number of tweets about Trump among Democrats was 97.12 (standard deviation = 115.27), but only 40.22 among Republicans (standard deviation = 44.81). In both cases, the mean was inflated by outliers with a large number of tweets (e.g., one Democrat authored 746 tweets about Trump and one Republican authored 288 tweets about Trump). Still,

the median number of tweets among Democrats (56) was still substantially larger than among Republicans (27). Only 8.6% of legislators had fewer than 5 tweets related to Trump.

The 10 Republicans with the most tweets about Trump were, in order: Paul Ryan Bradley Byrne, Sean Duffy, Paul Gosar, Bill Flores, Orrin Hatch, Mitch McConnell, Roger Wicker, Steve Scalise, Kevin McCarthy. The 10 Democrats with the most tweets about Trump were: Donald Beyer, Betty McCollum, Yvette Clarke, Jerrold Nadler, Edward Markey, James McGovern, Nancy Pelosi, Tom Udall, Robert Case, Joseph Crowley. In both cases we observe leadership in both parties among the most frequent authors of tweets about the president. Perhaps unsurprisingly, the days with both the most positive and the most negative tweets about Trump were those in which Trump addressed Congress: his joint address on February 28, 2017 (597 positive, 288 negative) and the 2018 State of the Union (512 positive and 340 negative).

Of the 29,696 Trump-related tweets from legislators, a subset of 4,661 tweets were randomly selected to be coded with respect to their sentiment about Trump. Five undergraduate research assistants were trained to categorize the sentiment of each tweet about Trump given the text of the tweet, the name of the legislator who sent it, and the legislator’s party affiliation. A random 1% sample of tweets was selected to be coded by each of the five coders in order to assess inter-coder reliability.⁸ The mean level of agreement in the coding of the tweets as positive, neutral, or negative was 91.7%.⁹ A table describing the percentage breakdowns for the labeled tweet sentiment classes according to party is provided in Table 5.

	% Positive	% Negative	% Neutral
Democratic	1.56	92.04	6.40
Republican	81.87	2.17	15.96

Table 5: Breakdown of labeled tweet sentiment classes according to party

See Section 3 for description of the splits into training, validation, and test datasets.

⁸After coding, we identified sixty-eight mislabeled tweets that were then corrected in the data set.

⁹The tweets were coded on a five-point scale (very positive, somewhat positive, neutral, somewhat negative, very negative), but the intercoder reliability was not sufficient to justify distinguishing “somewhat” from “very.” (91.7% vs. 60.0%).

B Model Selection Decisions

We examine our proposed model’s performance under different hyperparameters, including the dimensionality of the latent political space and the parameter γ that controls the tradeoff between the sentiment and count components of the model loss.

We select model hyperparameters based upon performance on a held-out validation set. We described in Section 3 the creation of our validation dataset. Using a validation set allows for evaluation of the model as it is developed, without exposing the model to the test set. This practice prohibits overfitting by ensuring a tuned model generalizes to wholly unseen data.

B.1 Tuning Model Dimension

We begin our selection of model hyperparameters with the model dimension: the dimension, K , of the political space of the legislator embeddings and Trump embeddings. In choosing this dimension, we fix all other attributes of the model and sweep through a range of possible model dimensions. For our dimension sweep, we fixed $\gamma = 0.01$. For each possible dimension, we fully train the model and obtain evaluative metrics – MAE on the labeled data, loss of the count model, and total loss – on the validation set. We then compare these metrics across dimension.

In Figure 4, we show the three evaluation metrics across of sweep of dimensions from 1 to 64. Between particularly the metrics of count loss and MAE, there is a trend of sharp decrease between dimension 1 and 2 and then a less discernible trend between dimension and metric for dimensions greater than 2. This indicates that across model dimension (greater than $K = 2$), performance with respect to our evaluative metrics remains relatively consistent. This allows the researcher a degree of flexibility in choosing model dimension. We further note that the evaluation metrics across dimension do not necessarily increase or decrease together. This further obfuscates the choice in model dimension, since the researcher may value optimizing a different metric depending on the chosen application. For the work presented in this paper – with predictive results and legislator embeddings shown in Sections 5 and 6, respectively – we selected a dimension of $K = 2$ to balance multiple research-defined objectives: to balance MAE, total loss, and count loss; to facilitate comparison to canonical DW-Nominate legislator representations;

to inhibit overfitting; and to allow for easy analysis. Furthermore, choosing $K = 2$ lends parsimony to our model without sacrificing performance across our evaluative metrics. We note that [Cranmer and Desmarais \(2017\)](#) discuss using predictive performance as an impartial means for choosing the parsimony of a model and refer interested readers to their discussion.

B.2 Tuning Loss Tradeoff Parameter γ

Similarly to tuning model dimension, K , we tune the loss tradeoff parameter γ by fixing all other attributes of the model and performing a sweep through a range of possible loss tradeoff values. We fixed the model dimension at $K = 2$. As with tuning model dimension, we fully train the model at each possible tradeoff value, and we again evaluate using the metrics of MAE, loss of the count model, and total loss on the validation set.

In Figure 5, we show the three evaluation metrics across of sweep of loss tradeoff values from 0.005 to 0.25. Unlike with model dimension, K , there is a discernible trend between evaluative metrics and the tradeoff parameter as it is swept. To balance the tradeoffs between the count model and sentiment model, we choose $\gamma = 0.03$ for the predictive results presented in our paper.

B.3 Comparing Model with Nonlinear Map

As mentioned in Section 4.3, when mapping from the text of Trump’s tweets to an embedding representation for Donald Trump, we may insert a nonlinear hidden layer to the model on top of an affine transformation. In this appendix, we compare the performance of the affine model against using a nonlinearity. The nonlinearity that we investigate is the rectified linear unit (ReLU).

Rather than comparing the affine and nonlinear models for only one dimension, we again perform a sweep over model dimension to investigate whether the superior model setting depends on region of the parameter space. In Figure 6, we show our three evaluation metrics, with a series for the model with and without the nonlinearity. The plots demonstrate that, in general, the affine model actually outperforms the nonlinear model. This is contrary to our initial expectation, since we would expect the nonlinear model to admit more flexibility, but given the results presented here, we use an affine model for our investigation in Sections 5 and 6.

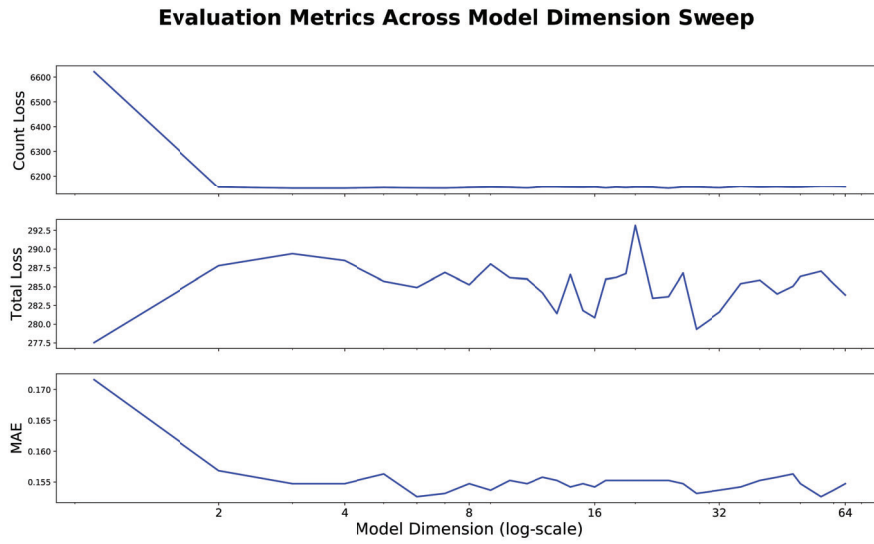


Figure 4: Evaluation metrics for the Basic model across a sweep of different model dimensions

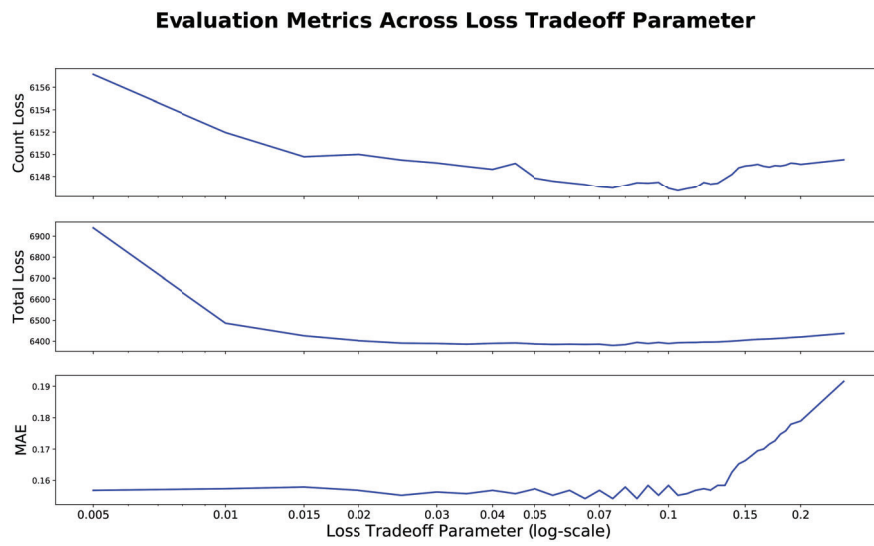


Figure 5: Evaluation metrics for the Basic model across a sweep of different loss tradeoff parameter values

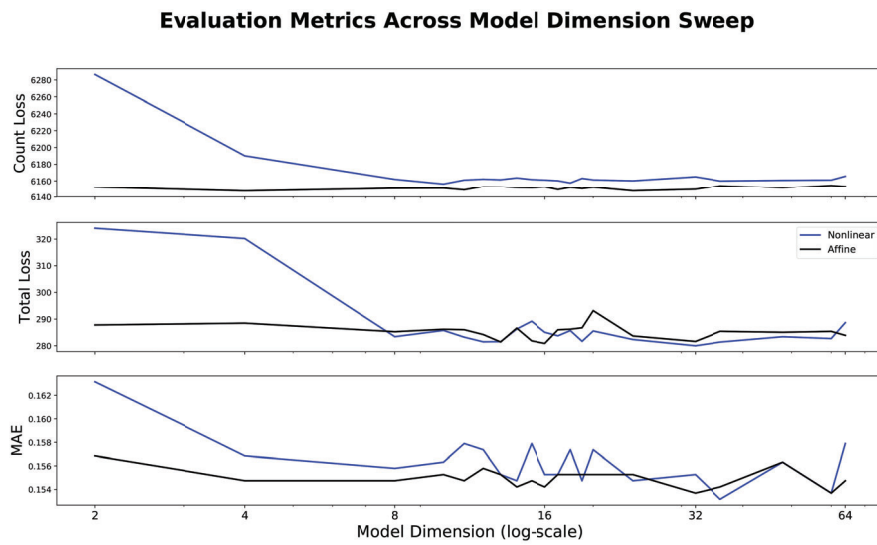


Figure 6: Evaluation metrics for the nonlinear and affine models over a sweep of model dimension

Comparison of Model Embeddings to Other Measures



Figure 7: Comparisons of the learned legislator embeddings to general measures of legislator preferences and proxies for a legislator’s support for Trump.

C Comparison with Other Preference Measures

In addition to the embedding comparisons provided in Section 6, we provide comparisons to Campaign Finance Scores [Bonica \(2018\)](#) and Trump’s vote margin in the 2016 presidential election for each legislator’s district or state, for representatives and senators, respectively. In the fourth panel of Figure 7 we observe a clear relationship between support for the president in the election and legislator embeddings—legislators representing constituencies that voted for Trump have lower embedding values.

D Legislator Embeddings with Labels

We reproduce the plot of our model-learned embeddings from Section 6 with explicit labels for Republican senators whom a 2017 *Washington Post* analysis identified as critical of the President based on their responses to controversial events in Trump’s presidency, such as Trump’s firing of FBI Director James Comey and response to the Charlottesville protests, as well as overall rhetoric used when discussing Trump ([Lewis et al., 2017](#)). This is presented in Figure 8.

Legislator Embeddings

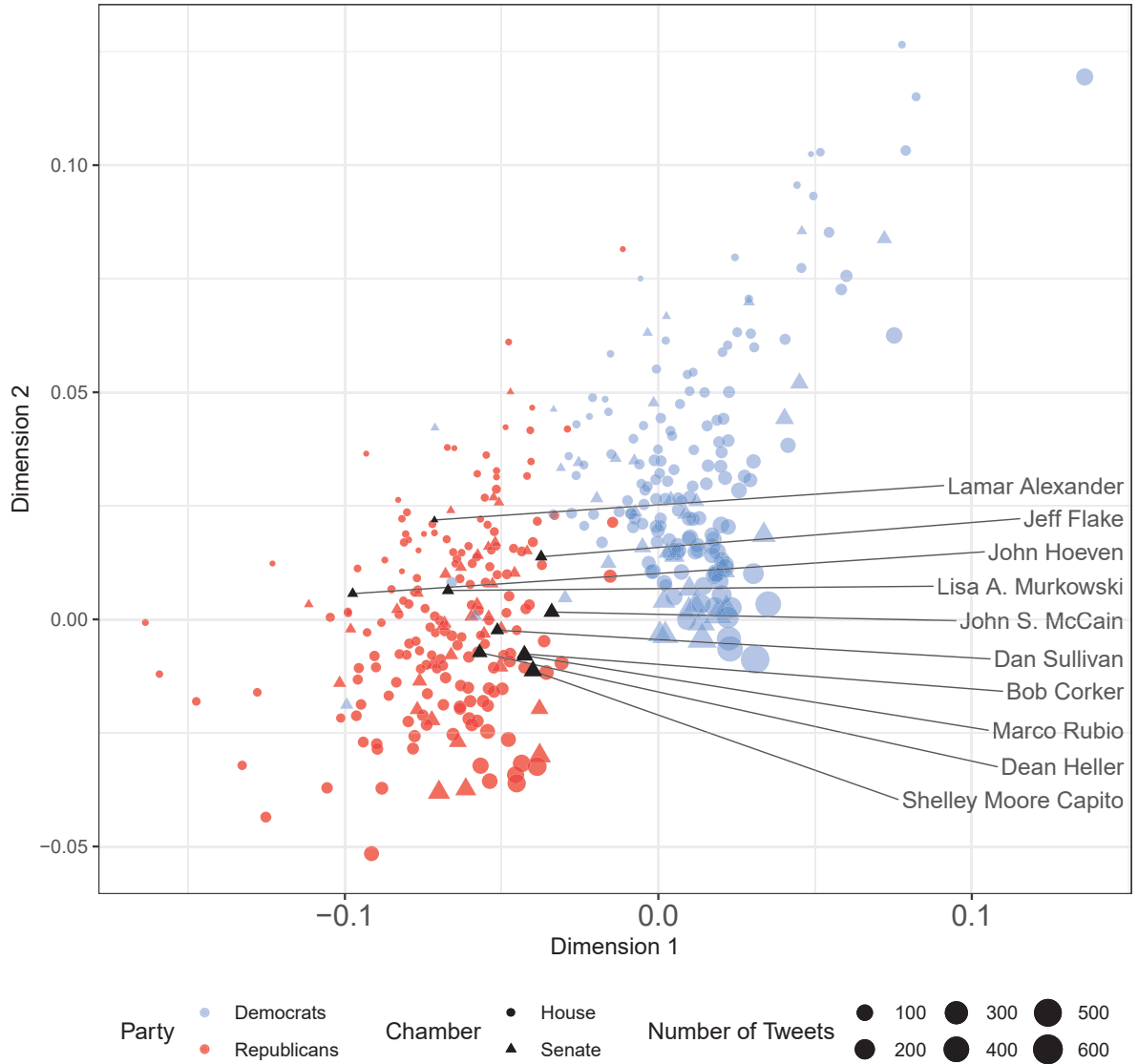


Figure 8: The two dimensions of learned legislator embeddings. Legislators are identified by party, chamber, and the number of tweets authored about Trump. The darker points represent the location of known Republican critics of Trump in the Senate.