An Embedding Model for Predicting Roll-Call Votes

Peter E. Kraft   Hirsh Jain   Alexander M. Rush
School of Engineering and Applied Science, Harvard University
{pkraft, hirshjain}@college.harvard.edu, srush@seas.harvard.edu

Abstract

We develop a novel embedding-based model for predicting legislative roll-call votes from bill text. The model introduces multidimensional *ideal vectors* for legislators as an alternative to single dimensional ideal point models for quantitatively analyzing roll-call data. These vectors are learned to correspond with pre-trained word embeddings which allows us to analyze which features in a bill text are most predictive of political support. Our model is quite simple, while at the same time allowing us to successfully predict legislator votes on specific bills with higher accuracy than past methods.

1   Introduction

Quantitative analysis of political data can contribute to our understanding of governments. One important source of such data is *roll-call votes*, records of how legislators vote on bills. Analysis of roll-call data can reveal interesting information about legislators (such as political leanings and ideological clusters) and can also allow prediction of future votes (Clinton, 2012).

Previous work on analyzing roll-call votes has chiefly involved positioning congresspeople on ideal point models. Ideal point models assume all legislators and bills can be plotted as single points in one-dimensional "political space." The closer a particular bill's position is to a particular congressperson's, the more utility the congressperson is expected to derive from the bill. Initial work on ideal point models focused on using them to test theories about legislative behavior, such as predicting that the relative differences between ideal points of congresspeople of different parties, and thus party polarization, would increase over time (McCarty, 2001). Ideal point models are often created using Bayesian techniques over large amounts of roll-call data (Clinton et al., 2004; Jackman, 2001). However, these models are not used to make predictions. They are trained using the complete vote matrix for the bill, which indicates how each congressperson voted on each bill. Therefore, they cannot say anything about how congresspeople will vote on a new bill, as until some congresspeople have voted on the bill its ideal point is not known.

We target this vote prediction problem: given the text of a bill and a congressperson, can we independently predict how each congressperson will vote on the bill? The first prior attempt at this task was made by Gerrish and Blei (2011) who create an *ideal point topic model* which integrates a topic model similar to LDA for the bill text with an ideal point model for the congresspeople. They use variational inference to approximate the posterior distribution of the topics and ideal points, predicting with a linear model. Gerrish and Blei (2012) further extend this work with an *issue-adjusted model*, a similar model that modifies congressperson ideal points based on topics identified with labeled LDA, but which cannot be used for predictions. Further work in a similar vein includes Wang et al. (2013), who introduced temporal information to a graphical model for predicting Congressional votes, and Kim et al. (2014), who used sparse factor analysis to estimate Senatorial ideal points from bill text and the votes of party leadership.

In this work we revisit this task with a simple bilinear model that learns multidimensional embeddings for both legislators and bills, combining them

to make vote predictions. We represent a bill as the average of its word embeddings. We represent legislators as *ideal vectors*, trained end-to-end for vote prediction. These ideal vectors serve as a useful, easy-to-train, multidimensional representation of legislator ideology that does not rely on elaborate statistical models or any further assumptions about legislator behavior. Finally, we train our model by optimizing a cross-entropy objective instead of the posterior of a topic model. The final model achieves high accuracy at predicting roll-call votes.

2 Model

Our goal is to predict roll-call votes by learning from the texts of bills and from past votes. Our input consists of a congressperson c and the set $\mathcal{B}$ of unique words in a bill. Our output y is whether that the congressperson voted yea or nay on the bill. We train on the full set of congressional votes on a number of bills. At test time, we supply entirely new bills and predict how each congressperson will vote on each new bill.

We propose a simple bilinear model that uses low-dimensional embeddings to model each word in our dictionary and each congressperson. We represent each bill using its word embeddings in order to capture the multivariate relationships between words and their meanings (Collobert et al., 2011; Mikolov et al., 2013). The model is trained to synthesize information about each congressperson's voting record into a multidimensional ideal vector. At test time, the model combines the embedding representation of a new bill with the trained ideal vector of a congressperson and generates a prediction for how the congressperson will vote on the bill.

Let $\mathbf{e}_w \in \mathbb{R}^{d_{word}}$ be the pretrained embedding for a word w. We initialize to the GloVe embeddings with $d_{word} = 50$ (Pennington et al., 2014), then jointly train them with the model. To represent a bill, we average over the embeddings of the set $\mathcal{B}$ of words in the bill.

To represent a congressperson, we introduce another set of embeddings $\mathbf{v}_c \in \mathbb{R}^{d_{emb}}$ for each congressperson c. The embeddings act as the ideal vector for each legislator. Unlike the word embeddings, we initialize these randomly.

The full model takes in a bill and a congressper-

Congress	# Bills	House	Senate	Pres
106	557	R	R	Clinton
107	505	R	D[2]	Bush
108	607	R	R	Bush
109	579	R	R	Bush
110	854	D	D	Bush
111	965	D	D	Obama

Table 1: Dataset details for 106-111th Congress.

son. It applies an affine transformation, represented by a matrix $\mathbf{W} \in \mathbb{R}^{d_{emb} \times d_{word}}$ and bias $\mathbf{b} \in \mathbb{R}^{d_{emb}}$, to map the bill representation into the space of the ideal vectors, and then uses a dot-product to provide a yea/nay score.

$$p(y = yea|\mathcal{B}, c) = \sigma((\mathbf{W}\left(\sum_{w \in \mathcal{B}} \mathbf{e}_w/|\mathcal{B}|\right) + \mathbf{b}) \cdot \mathbf{v}_c)$$

The full model is simply trained to minimize the negative log-likelihood of the training set, and requires no additional meta-information (such as party affiliation) or additional preprocessing of the bills during training- or test-time.

3 Experimental Setup

Data Following past work, our dataset is derived from the Govtrack database.[1] Specifically, our dataset consists of all votes on the full-text (not amendments) of bills or resolutions from the 106th-111th Congress, six of the most recent Congresses for which bill texts are readily available. Details of each these congresses are shown in Table 1.

To create our dataset, we first find a list of all votes on the full text of bills, and create a matrix of how each congressperson voted on each bill, which will be used in training and in testing. In accordance with previous work, we only consider yes-or-no votes and omit abstentions and "present" votes (Gerrish and Blei, 2011). We then simply collect the set of words used in each bill. Overall, our dataset consists of 4067 bills and over a million unique yes-or-no votes.

---

[1]https://www.govtrack.us/

[2]Mostly. Republicans controlled the 107th Senate for five months between the inauguration of Dick Cheney as vice-president in January of 2001 and the defection of Jim Jeffords in June.

Congress	YEA	GB	IDP	EMB
106	83.0	-	79.5	84.9
107	85.9	-	85.8	89.7
108	87.1	-	85.0	91.9
109	83.5	-	81.5	88.4
110	82.7	-	80.8	92.1
111	86.0	-	85.7	93.4
Avg	84.5	89.0	83.1	90.6

Table 2: Main results comparing predictive accuracy of our model EMB with a several baselines (described in the text) on the 106th-111th Congress.

Congress	EMB
106	0.524
107	0.546
108	0.595
109	0.628
110	0.728
111	0.737
Avg	0.645

Table 3: Minority class F1 Scores of our model EMB on the 106th-111th Congress.

Model We tested prediction accuracy of the average-of-embeddings model, EMB, by running it for ten epochs at a learning rate of $\eta = 0.1$ and d_{emb} set to 10. Hyperparameters were tuned on a held-out section of the 107th Congress. We ran on each of the 106th to 111th Congresses individually using five-fold cross-validation.

Baselines We compare our results to three different baselines. The first, YEA, is a majority class baseline which assumes all legislators vote yea. The second, IDP, is our model with d_{emb} set to 1 to simulate a simple ideal point model. The third, GB, is Gerrish and Blei's reported predictive accuracy of 89 % on average from the 106th to 111th Congresses, which is to the extent of our knowledge the best predictive accuracy on roll-call votes yet achieved in the literature. Gerrish and Blei report on the same data set using cross-validation and like us train and test on each congress individually, but do not split out results into individual congresses.

4 Experiments and Analysis

Predictive Results The main predictive experimental results are shown in Table 2. We see that EMB performs substantially better than YEA on all six Congresses. It has a weighted average of 90.6% on an 84.5% baseline, compared to Gerrish and Blei's 89% on an identical dataset. IDP, however, actually does worse than the baseline, demonstrating that the bulk of our gain in prediction accuracy comes from using ideal vectors instead of ideal points. To further test this hypothesis, we experimented with replacing word embeddings with LDA

and obtained an accuracy of 89.5%, in between GB and EMB. This indicates that the word embeddings are also responsible for part, but not all, of the accuracy improvement. We also report minority class F1 scores for EMB in Table 3, finding an overall average F1 score of 0.645.

Ideal Vectors Beyond predictive accuracy, one of the most interesting features of the model is that it produces ideal vectors as its complete representation of congresspeople. These vectors are much easier to compute than standard ideal points, which require relatively complex and computationally intensive statistical models (Jackman, 2001). Additionally unlike ideal point models, which tend to contain many assumptions about legislative behavior, ideal vectors arise naturally from raw data and bill text (Clinton et al., 2004).

In Figure 1, we show the ideal vectors for the 111th Congress. We use PCA to project the vectors down to two dimensions. This graph displays several interesting patterns in agreement with theories of legislative behavior. For example, political scientists theorize that the majority party in a legislature will display more unity in roll-call votes because they decide what gets voted on and only allow a vote on a bill if they can unify behind it and pass it, while that bill may divide the other party (Carrubba et al., 2006; Carrubba et al., 2008). On this graph, in accordance with that prediction, the majority Democrats are more clustered than the minority Republicans. We observe similar trends in the ideal vectors of the other Congresses. Moreover, the model lets us examine the positions of individual congresspeople. In the figure, the 34 Democrats who

Figure 1: PCA projection of the ideal vectors for 111th Congress, both House and Senate. Republicans shown in red, Democrats who voted for Affordable Care Act (ACA) in blue, Democrats who voted against ACA in yellow, and independents in green.

Figure 2: Relative likelihood of congresspeople in the 110th Congress voting for a bill containing only the word "Enterprise" versus only the word "Science." Coordinates are sigmoids of dot products of congressperson vectors with normalized word vectors.

voted against the Affordable Care Act (ACA, better known as Obamacare) are shown in yellow. The ACA was a major Democratic priority and point of difference between the two parties. The Democrats who voted against it tended to be relatively conservative and closer to the Republicans. The model picks up on this distinction.

Furthermore, since our model maps individual words and congresspeople to the same vector space, we can use it to determine how words (and by proxy issues) unite or divide congresspeople and parties. In Figure 2, we show the scaled probabilities that congresspeople will vote for a bill containing only the word "enterprise" versus one containing only the word "science" in the 110th Congress. The word "enterprise," denoting pro-business legislation, neatly divides the parties. Both are for it, but Republicans favor it more. More interestingly, the word "science" creates division *within* the parties, as neither was at the time more for science funding than the other but both contained congresspeople with varying levels of support for it. An ideal point model would likely capture the "enterprise" dimension, but not the "science" one, and would not be able to distinguish between libertarians like Ron Paul (R-TX) who are against both "corporate welfare" and government science funding, conservative budget hawks like Jeff Flake (R-AZ) who favor business but are skeptical of government funding of science, and es-
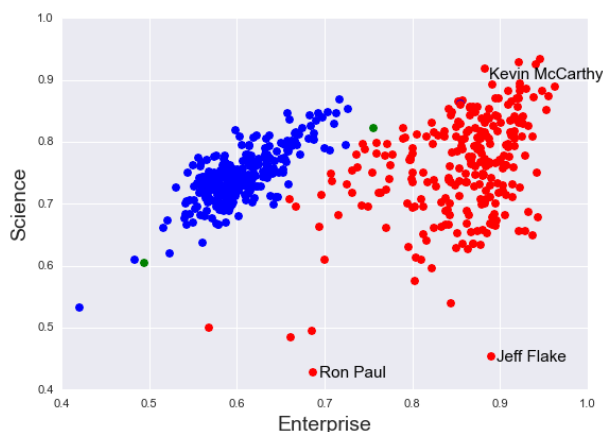
tablishment Republicans like Kevin McCarthy (R-CA) who support both. Indeed, ideal point models are known to perform poorly at describing ideologically idiosyncratic figures like Ron Paul (Gerrish and Blei, 2011). Providing the ability to explore multiple dimensions of difference between legislators will be extremely helpful for political scientists analyzing the dynamics of legislatures.

Lexical Properties   Finally, as with topic modeling approaches, we can use our model to analyze the relationships between congresspeople or parties and individual words in bills. For example, Table 4 shows the ten words closest by cosine similarity to each party's average congressperson (stop words omitted) for the 110th Congress. The Democratic list mostly contains words relating to governing and regulating, such as "consumer," "state," and "educational," likely because the Democrats were at the time the majority party with the responsibility for passing large governmental and regulatory bills like budgets. The Republican list is largely concerned with the military, with words like "veterans," "service," and "executive," probably because of the importance at the time of the wars in Iraq and Afghanistan, started by a Republican president.

Democrats	Republicans
economic	veterans
exchange	head
state	opportunities
carrying	provided
government	promote
higher	service
congress	identified
consumer	information
educational	record
special	executive

Table 4: Top ten words by cosine similarity for each party in the 110th Congress with stop words removed.

5 Conclusion

We have developed a novel model for predicting Congressional roll-call votes. This new model provides a new and interesting way of analyzing the behavior of parties and legislatures. It achieves predictive accuracies around 90.6% on average and outperforms any prior model of roll-call voting. We also introduce the idea of ideal vectors as a fast, simple, and multidimensional alternative to ideal point models for analyzing the actions of individual legislators and testing theories about their behavior. Our code and datasets are available online at https://github.com/kraftp/roll_call_predictor.

References

Clifford J Carrubba, Matthew Gabel, Lacey Murrah, Ryan Clough, Elizabeth Montgomery, and Rebecca Schambach. 2006. Off the record: Unrecorded legislative votes, selection bias and roll-call vote analysis. *British Journal of Political Science*, 36(04):691–704.

Clifford Carrubba, Matthew Gabel, and Simon Hug. 2008. Legislative voting behavior, seen and unseen: A theory of roll-call vote selection. *Legislative Studies Quarterly*, 33(4):543–572.

Joshua Clinton, Simon Jackman, and Douglas Rivers. 2004. The statistical analysis of roll call data. *American Political Science Review*, 98(02):355–370.

Joshua D Clinton. 2012. Using roll call estimates to test models of politics. *Annual Review of Political Science*, 15:79–99.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

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

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

Simon Jackman. 2001. Multidimensional analysis of roll call data via bayesian simulation: Identification, estimation, inference, and model checking. *Political Analysis*, 9(3):227–241.

In Song Kim, John Londregan, and Marc Ratkovic. 2014. Voting, speechmaking, and the dimensions of conflict in the us senate. In *Annual Meeting of the Midwest Political Science Association*.

Nolan McCarty. 2001. The hunt for party discipline in congress. In *American Political Science Association*, volume 95, pages 673–687. Cambridge Univ Press.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

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