Legislator Arithmetic *

Daniel Argyle *FiscalNote*

We introduce a neural network implementation of ideal point estimation called NN-NOMINATE that scales well to large datasets and allows incorporation of additional metadata. We test the model on synthetic data, the entire corpus of US Congressional votes, and cosponsorship decisions since the 93rd Congress. These experiments show that the model behaves as expected and that, when using a held out set of votes as a validation set, we are able to make informed decisions about the model structure including the number of dimesions and whether or not to add covariates.

Keywords: ideal point estimation

Introduction

We propose a neural network implementation of ideal-point estimation that scales well to large datasets and allows incorporation of additional metadata, which we call NN-NOMINATE. Neural networks are well suited for these models, and the performance benefit, along with distributed computing capabilities, allows application of ideal point estimation to pooled datasets where computation was previously infeasible due to scale. We demonstrate the algorithm on two different datasets, the complete history of US Congressional roll call votes and modern cosponsorship behavior, and compare the results against standard ideal point estimation techniques.

To evaluate algorithmic performance, we test the resulting estimates on both training and test data by holding out a subset of legislators' votes. This allows us to compare the quality of different model parameterizations and choice of dimensions while still guarding against overfitting. Specifically, we directly compare the performance of NN-NOMINATE to different ideal point parameterizations such as DW-NOMINATE and the conventional Bayesian parameterization.

We demonstrate NN-NOMINATE in two ways. First, we jointly estimate ideal points over the pooled set of US Congressional roll call votes from 1789-2018. Unidimensional ideal points from the neural network implementation are similar to the conventional DW-NOMINATE results. However, cross validation scores indicate that the data are better explained with more than one dimension. Clustering the multidimensional ideal points yields intuitive temporal and ideological groupings and provides a more nuanced picture of ideological polarization.

Second, we take advantage of the fact that many more bills are sponsored than actually come to a vote and estimate an ideal point distribution over a large set of sponsorship and cosponsorship decisions in the 93rd-114th Congresses. Cosponsorship provides a different perspective on legislators' beliefs, independent of strategic voting or administrative votes of little ideological salience. We treat cosponsorship as a clear endorsement of a bill's content and assume that a choice not to cosponsor a bill can be interpreted as something less than full support. When compared to traditional ideal points, cosponsorship ideal points show somewhat different trends in polarization and result in a higher number of optimal dimensions.

*The code for this method is available at the author's github: https://github.com/dargyle/legislator-arithmetic

Implementing Ideal Points as a Neural Network

This work is inspired by two similar lines of research in political science and computer science. Speaking broadly[1], political scientists – from the the seminal work of Poole and Rosenthal (1985) and Clinton, Jackman and Rivers (2004), and up to and including modern contributions such as Imai, Lo and Olmsted (2016) – have focused primarily on ideal point estimation as a means to position legislators within the ideology space. That these methods also predict votes is somewhat of an afterthought. On the other hand, computer science implementations largely focus on using an ideal point framework to predict legislator votes, without interpreting the ideal points themselves (for example see Gerrish and Blei (2011); Kraft, Jain and Rush (2016)).

Combining the insights of these two fields, we wish to use the prediction-focused framework of computer science to select the most accurate model, which in turn would provide the most informative ideal point estimations for substantive interpretation. We suggest that 1) the model that predicts the best *on a held out sample of votes* provides the most insight into ideal points and 2) there is a clear trade-off between explanatory power and ease of interpretation of ideal point models that should be made explicit.[2]

Model Frameworks

There are two broad frameworks that have been commonly used for ideal point estimation. Both tie back to spatial voting theory and the idea that a legislator will prefer something that is "near" to them in the ideology space. This ideology space can be paramterized with an aribtrary number of dimensions (denoted with K), but in most applications K is set to be 1 or 2.

The first framework, the NOMINATE family, posits that the probability that a legislator votes yes on a bill is given by:

$$prob(vote = yea) = \Phi\left(u_{ijy} - u_{ijn}\right) = \Phi\left(\beta\left[\exp\left(-\frac{1}{2}\sum_{k=1}^{s} w_k^2 d_{ijyk}^2\right) - \exp\left(-\frac{1}{2}\sum_{k=1}^{s} w_k^2 d_{ijnk}^2\right)\right]\right) \tag{1}$$

where u_{ijy} is the utility legislator i receives from voting yes (y) on proposal j.[3] This utility is expressed in terms of the squared distance d_{ijyk}^2 between a legislator's ideal point and the "yes" outcome point (and similarly d_{ijnk}^2 is the distance between the legislator's ideal point and the "no" outcome point). There are also a set of salience weights w_k^2 which allow different dimensions to have more impact on the final voting decision. The function Φ is commonly a normal or logistic cdf, which makes this model, once the weights have been obtained, essentially a probit or logit regression with a single parameter β.

The second framework, the item-response family, commonly associated with Clinton, Jackman and Rivers (2004), predicts a vote as follows:

$$prob(vote = yea) = \Phi\left(\beta_j \cdot \mathbf{x}_i + \alpha_j\right) \tag{2}$$

[1] A complete review of the literature in either field is beyond the scope of this work, and there are, unsurprisingly, some exceptions to this assertion.

[2] We will provide evidence that the optimal number of dimensions for ideal point estimation is larger than the most common one or two. It is perfectly reasonable to choose to rely on two dimensions, but the trade-offs of that choice should be clear.

[3] For simplicity, we are surpressing the time dimension that is often present in these models. However, NN-NOMINATE generalizes to include dynamic ideal points as is discussed later in the paper. See also Carroll et al. (2009) for more detail about the background of this model.

where $\mathbf{x}_i$ is a K dimensional vector of ideal points and again Φ can correspond to either a logistic or probit model. The bill level parameter β_j, sometimes refered to as polarity, is multiplied with the ideal point vector and determines the lean of the bill. The parameter α_j, sometimes referred to as popularity, determines a baseline approval rate for a specific proposal.

The Neural Network

While neural networks have become most strongly associated with artificial intelligence, they are, in essence, flexible optimization systems fitted using various forms of gradient decent algorithms.[4] All that is required to estimate an ideal point model using a neural network is to set the network structure to match the underlying framework of the ideal point model. One example can be seen in Figure 1, which represents the architecture of a W-NOMINATE model. We will walk through each layer of this example in some detail as a baseline for further exploration.

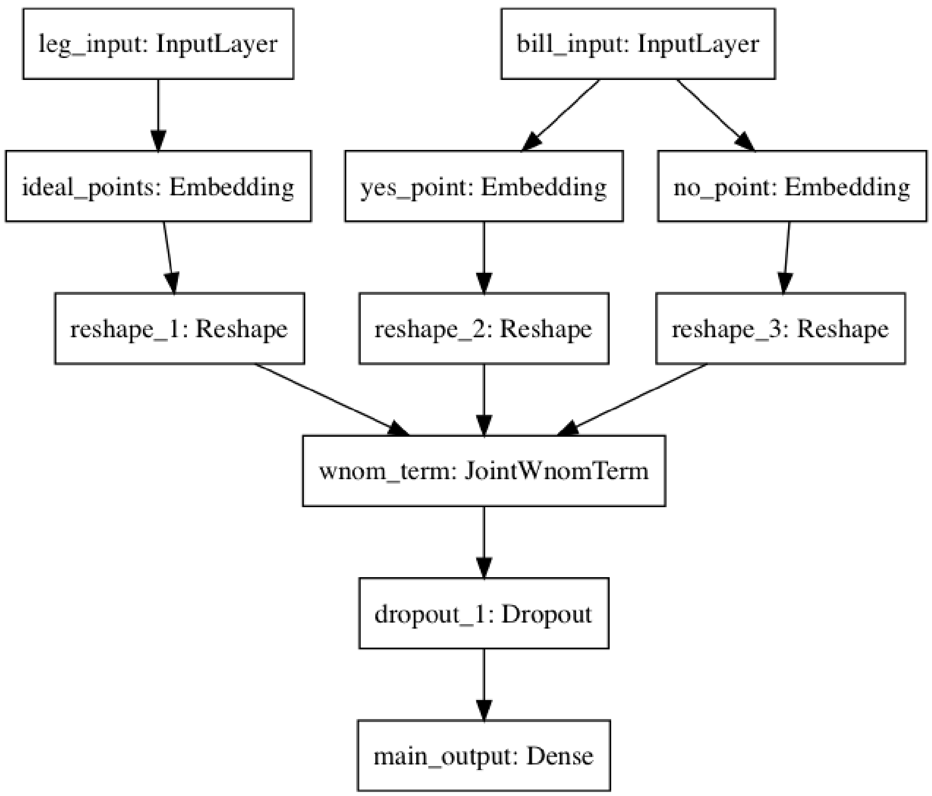

Figure 1: Base Model

- Input layers: The inputs to this model consist of numeric identifiers for both legisators and bills.

- Embedding layers: These layers take the numeric identifiers and convert them into continuous parameters, for example mapping a specific legislator to her ideal point vector. In this model there is one ideal point embedding for legislators, and two embeddings for

[4] Indeed, while not directly applicable to the models contained here, there are several papers proving that feed forward neural networks can approximate any continuous function (**?**)cybenko1989approximation, hornik1991)

bills, the yes point and the no point, that correspond to numeric values associated with W-NOMINATE style estimation. The ideal point layer is restricted in two ways. First, the model includes orthogonality regularization on the ideal points (Brock et al., 2016; Vorontsov et al., 2017). This penalizes correlation between ideal point dimensions and ensures that the resulting estimates are orthogonal. Ideal point embeddings also have a maximum norm constraint, which ensures that the norm of the ideal point vector for any legislator lies within the unit hypersphere.[5]

- Reshape layers: Because they are often used for text analysis, by default embedding layers assume that there is a sequence of ids to to transform into the embedding space. Since we have only a single id to embed, these layers drop the dimension associated with the sequence to form a two dimensional tensor.

- JointWnomTerm layer: The embedded and reshaped data is fed into a custom neural network layer that implements the conventional WNOMINATE estimation (as found in Poole et al. (2011)). Specifically, the layer implements the inner portion of model in 1, sepcifically calculating

$$\exp\left(-\frac{1}{2}\sum_{k=1}^{s} w_k^2 d_{ijyk}^2\right) - \exp\left(-\frac{1}{2}\sum_{k=1}^{s} w_k^2 d_{ijnk}^2\right) \tag{3}$$

  where K is the number of dimensions in the model, w_k is the salience weight for dimension k, and d_{iyk}^2 is the distance between a legislator's ideal point and the yes point of the bill and d_{ink}^2 is the distance between the legislator's ideal point and the no point of the bill.[^10]

- Dropout layer: Dropout regularization, as proposed by Srivastava et al. (2014), has become an extremely common way of limiting model overfitting, often with the additional benefit of improved behavior during model training. Dropout layers set model weights to 0 at random during iterations of the training process. This prevents any single weight dominating the prediction of the algorithm. In this specific instance, the dropout layer sets the salience weight for any given ideology dimension to 0 for a specific iteration of the optimzation algorithm relying upon the other dimensions to make a vote prediction. In practice, this has resulted in better model performance in terms of observable metrics and by limiting overfitting during the training process.

- Dense layer: The quantity obtained in the JointWnomLayer is then used as an input to a Dense layer, parameterised as a logistic regression (i.e. sigmoid activation and binary_-crossentropy loss). The single model weight estimated here is denoted β in most NOMINATE models and represents the signal to noise ratio. A high value corresponds to votes being largely deterministic, a lower value suggest more randomness.

The data is given as a batch of triples, $(legislator\_id, bill\_id, vote)$. The identifiers are fed into the input layers and subsequently into the embeddings to generate the predictive parameters of the network. The vote itself is used as the outcome value of the final model layer (the logistic parameterized Dense layer).

[5]Note that the maximum norm constraint does not require that any legislator actually attain a unit norm. This differs somewhat from existing implementations which seem to ensure all dimensions fill the range from [-1, 1]. If this behavior is desired, it can be easily fixed in postprocessing.

The Neural Network Implementation Works

We demonstrate that the neural network implementation works at least as well as existing methods through two simple tests. The first is to test the model on synthetic data, generated with known parameters, and determine how well the method recovers these known values. The second is to check that the results match the commonly used implementations of W-NOMINATE (Poole et al., 2011) and item response (Jackman, 2017) ideal points.

Recovering Known Ideal Points

The synthetic data is generated to have 3 known dimensions, where the random predictions are generated in the NOMINATE parameterization. To guard against overfitting to specific votes a legislator took, we train the model on a subset of the data and test it on 20% of votes as a held out set. This means that a legislator's ideal point is determined only by the votes in the training sample and that a bill's paramters are determined only by the subset of legislators included who voted on this bill.

There are two primary benefits to using a held out sample. First, relying on an out of sample evaluation set is often used to determine when a neural network has converged while estimating a model. There is little point in continuing to optimize a model where out of sample performance has plateaued and doing so usually results in overfitting.

Second, performance on a held out sample can be used to make additional decisions about the model structure, including parameterization choices or for evaluating distributional assumptions.[6] For example, how does making the ideal points dynamic affect the model performance? What is the optimal number of ideal point dimensions? An additional dimension may offer an improvement in out of sample prediction, but it almost always increases in sample performance. An out of sample performance test provides an objective metric to determine the optimal number of dimensions to include.[7] If a practitioner finds it undesirable to omit votes in this fashion, the model could then be refit on the entire dataset using the model setup that performed best on the test data.

Figure 2 shows the model performance of NN-NOMINATE on synthetic data, varying the number of dimensions in the model. We evaluate two metrics, accuracy (whether or not the vote was predicted correctly) and log-loss (a common metric used in classification problems). Log-loss is useful because it provides an idea of how good the probability estimates of a model are, and penalizes over-confident predictions. We see that the log-loss decreases rapidly and is minimized at the optimal number of dimensions on the test data, following which the test log-loss begins to increase. Training accuracy shows similar patterns, attaining best performance on the known optimal number of dimensions.

Additionally, Figure 3 shows evidence that NN-NOMINATE obtains plausible estimates of the ideal point parameters in the proper three dimensions. It is important to note that the synthetic data was generated with the first dimension having a higher salience weight than the other two, which accounts for why the first dimension is more precisely correlated. Also, because the model was only fit on 80% of the data in training, some imprecision is expected. An equivalent chart on the full dataset yields more precise estimates.

[6] A recent paper by \cite{{marble2017much} uses a held out set to discuss the predictive power of ideal points in survey data, and in doing so estimate ideal points on Senate votes from the 111th-114th congress using a heldout set and cross-validation. To our knowledge, there have not been other attempts to use this out of sample validation strategy more broadly.

[7] It is not as simple as saying that "the model that fits best on test data" is the right answer. In some cases it is

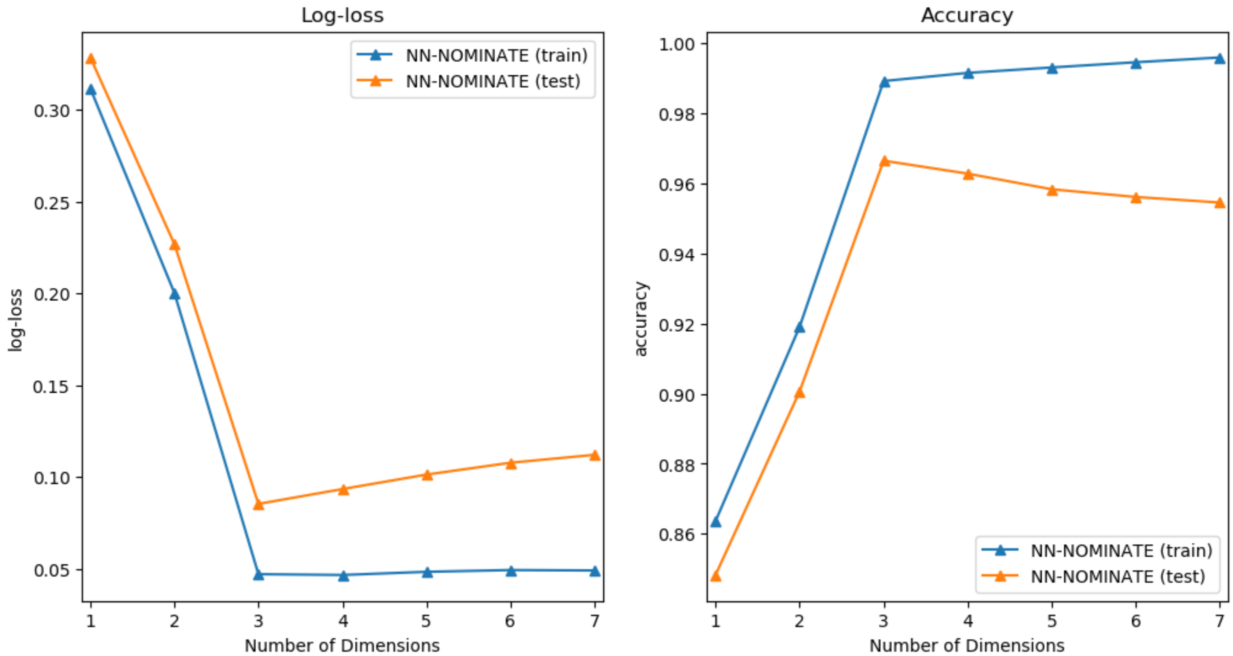

Figure 2: Performance metrics on Synthetic Data

Since is has not yet been common to estimate ideal point models on a training and test set, we verify that the model works by comparing the results of NN-NOMINATE to the WNOMINATE package available in R. The results for the synthetic data are shown in Figure 4, where WNOMINATE shows similar patterns over training and test data. It appears that for the synthetic dataset at least, NN-NOMINATE (slightly) outperforms the commonly used WNOMINATE package attaining lower log loss and higher accuracy on test data.

Comparisons on real data

In Figure 5, we show the same model evaluation metrics on real world data from the 110th-115th US Senate, which is roughly the same size as the synthetic test set. In this case there seems to be little evidence that there exists more than one dimension in the test data, as both NN-NOMINATE and WNOMINATE achieve minimum at the beginning (prior to NN-NOMINATE beginning to slightly overfit as the number of dimensions goes above 4).

We can also use simple correlations between the estimated ideal points to make sure that the ideal point space is similar across estimation methods. These results are shown in Figure **??** Since ideal points are not independently identified, including up to dimension switching and other problems, there are some important caveats. First, it is often the case that NN-NOMINATE results in ideal points that are smaller in maginitude on average than the WNOMINATE package and second, dimension labels can switch across the two different methods.[8]

preferable to say that this is the first model that attains a near minimum.

[8]Indeed labels often switch across runs of the same model. The exception for NN-NOMINATE, which because of initialization choices (we impose initial values for Republicans in the sample on ther right and Democrats on the left), almost always results in the first dimension being the most dimesion most strongly related to party. Note that the predictive results do not depend on this initialization choice, and that it is merely selected for convenience.

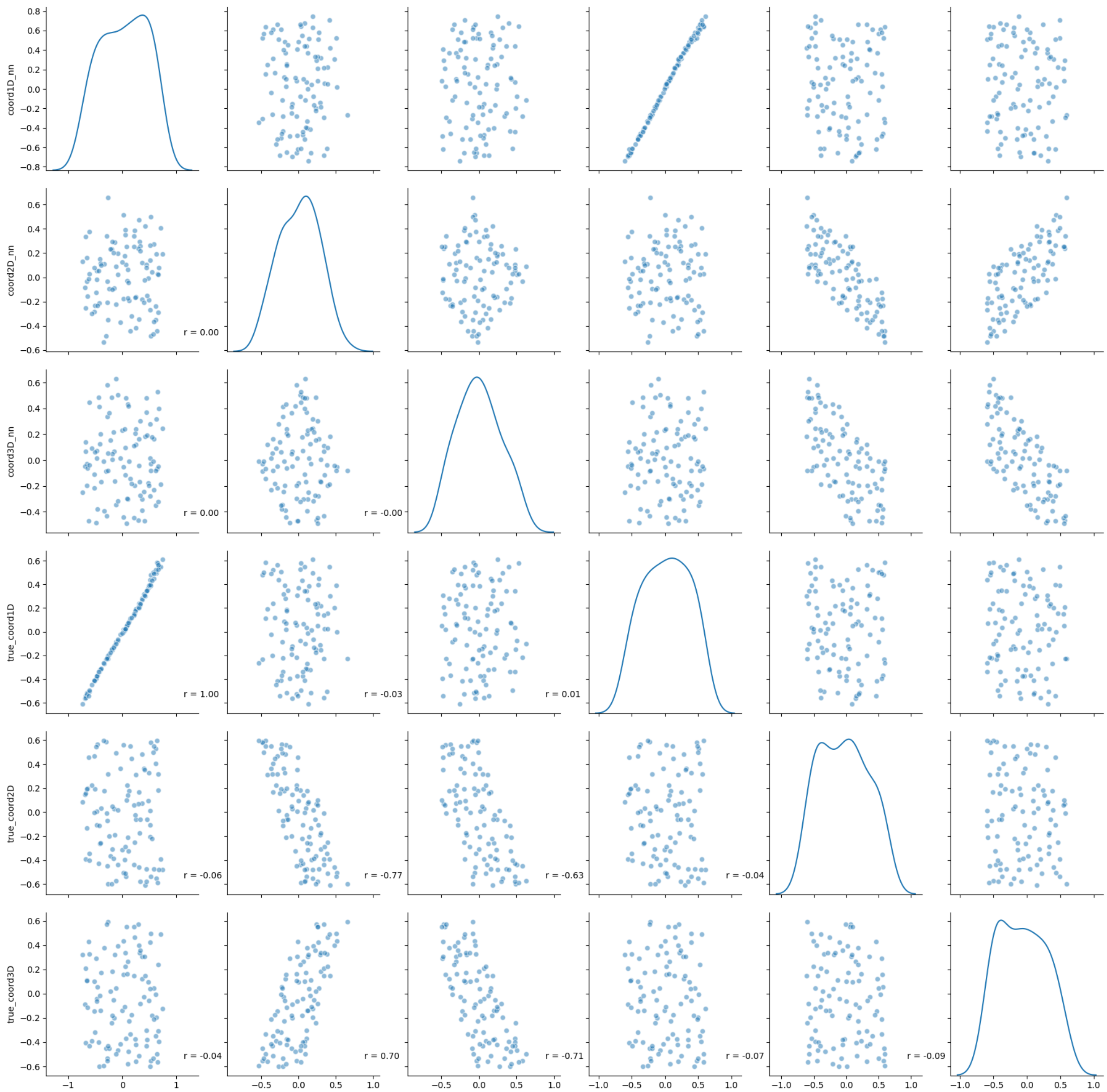

Figure 3: NN-NOMINATE and "True" ideal point comparison

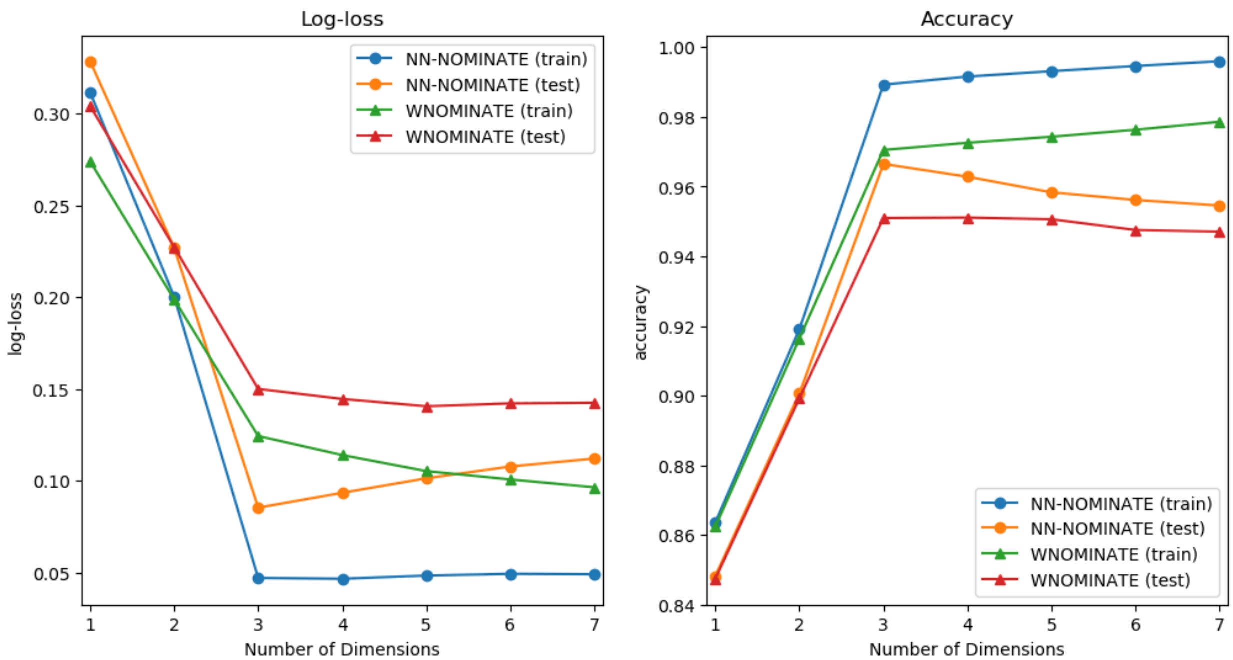

Figure 4: Performance metrics on Synthetic Data with WONOMINATE

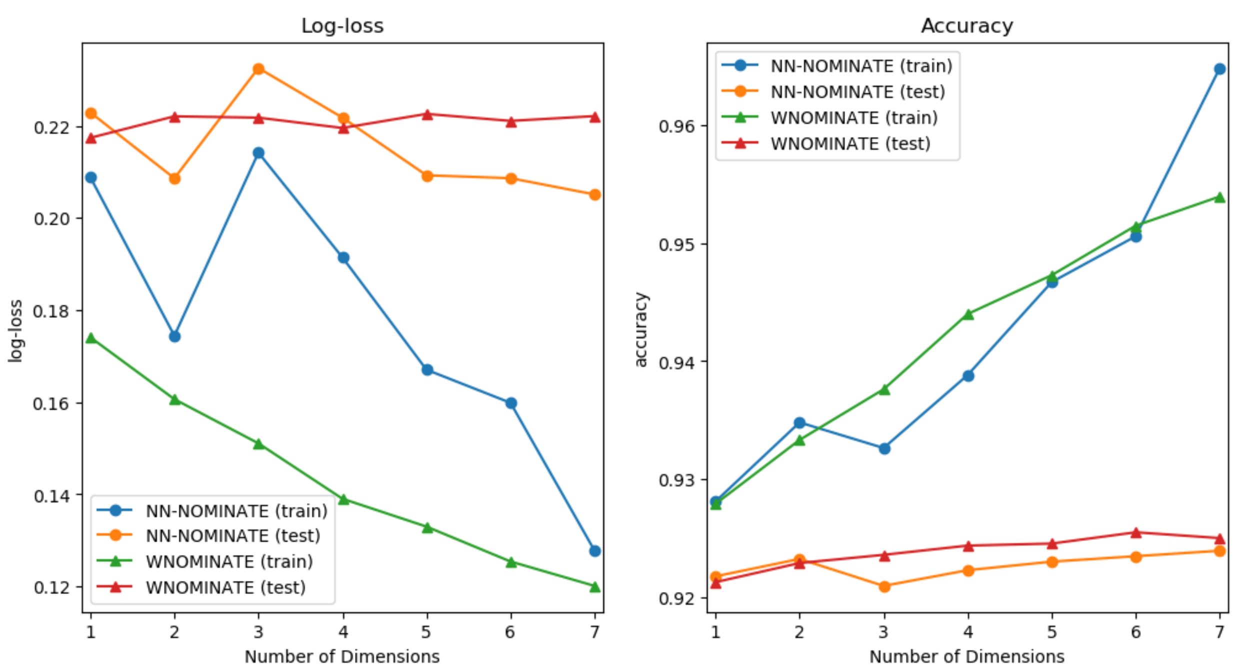

Figure 5: Model metrics on 110th-115th Senate

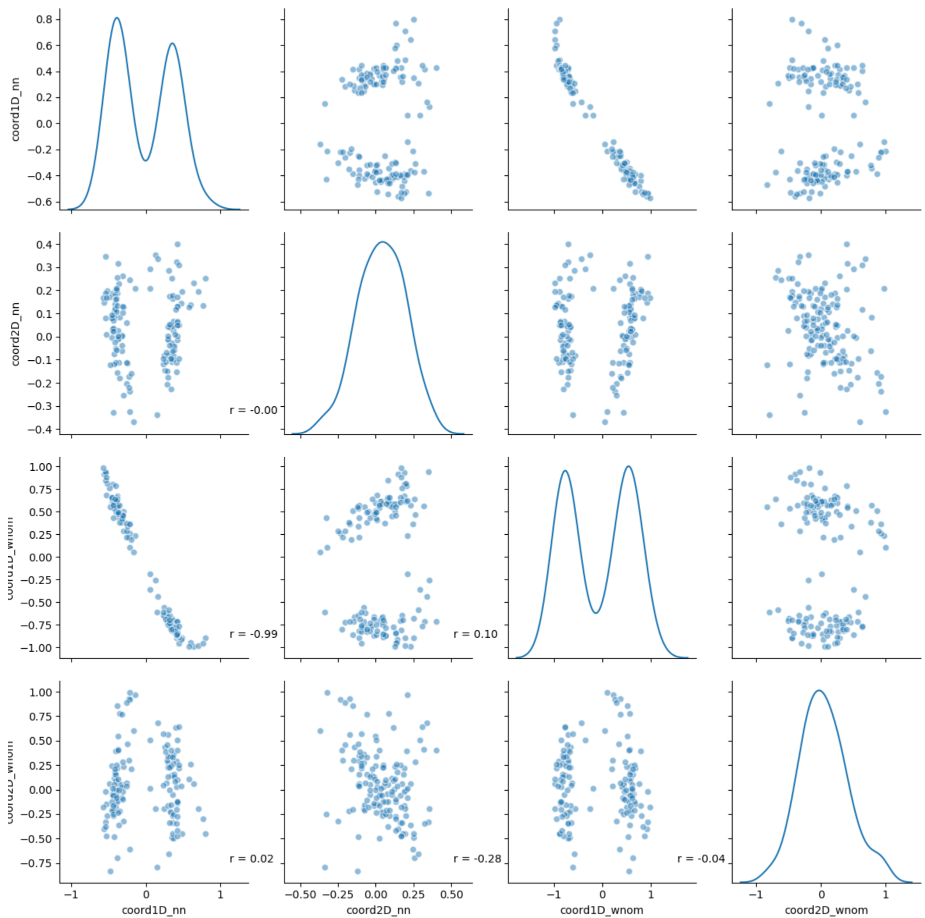

Figure 6: NN-NOMINATE and WNOMINATE estimates comparison on 110th-115th Senates

Results and applications

While it is comforting to know that a new implementation of an existing model returns similar results, NN-NOMINATE is capable of more than the existing implementations. The first is large scale ideal point estimation, for example fitting models on the entire history of the US Congress locally on a laptop. The second is to demonstrate the extensibility of the framework, where the model can be extended very simply to include covariates as well as more complicated representations of a bill than existing representations.

Large scale implementation

We examine two large datasets, all of which can be fitted easily on most common computers.[9] The first is simply the entire history of US rollcalls. The dimension tradeoff calculation from above is shown in Figure 8, which shows that the optimal number of dimensions is at least 8 over this entire dataset.

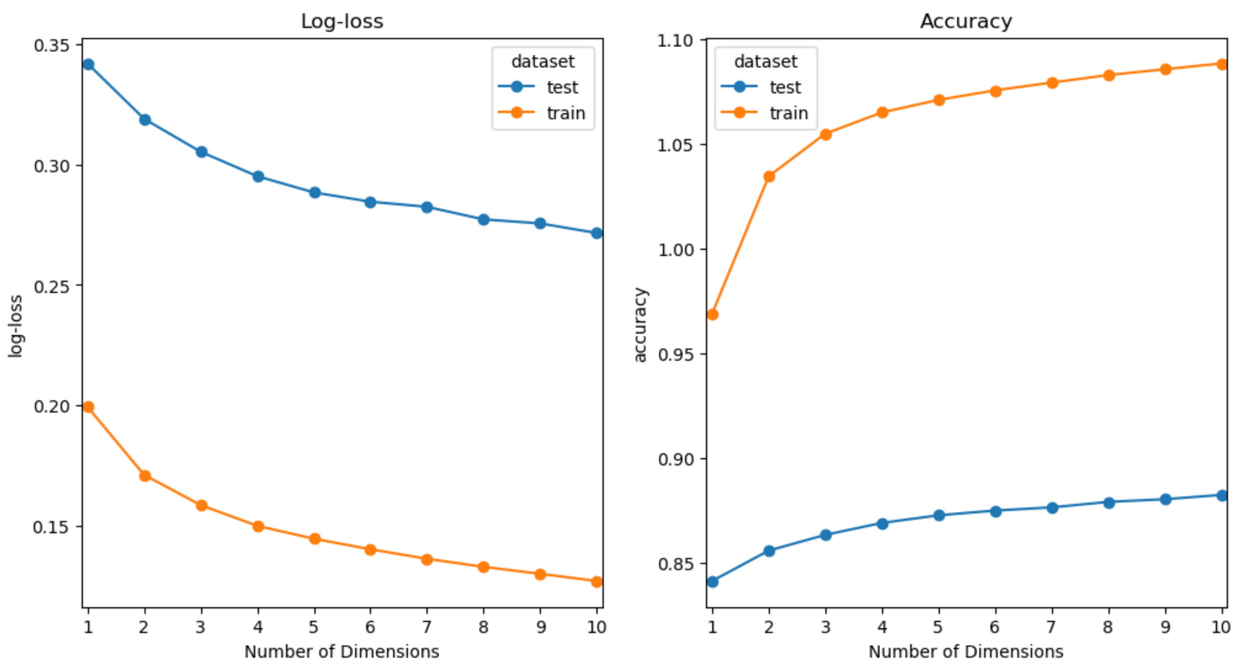

Figure 7: Metrics on Ideal-Points from the 1st-115th Congresses

The second dataset is the set of all cosponsorship decisions from the 93rd-114th US Congresses. This ideal at least goes back to Talbert and Potoski (2002)[10], who find that there are more dimensions in cosponsorship decisions than in voting. Since so many voting decisions include strategic concerns, cosponsorhip, a potentially less impactful choice, could allow legislators to espouse a wider variety of policy opinions (and thus require additional dimensions in the modeling process).

In this case, a "yes" vote is considered to be sponsoring or cosponsoring a bill, whereas everyone who did not is assumed (at least by opportunity cost) to be a "no". Note that this data is

[9] Neural networks have reputations as require lots of processing power, including the use of graphics cards. However, for a straightforward model like this, such processing power is not necessary. Formal performance comparisions to existing methods is forthcoming.

[10] See also Alemán et al. (2009).

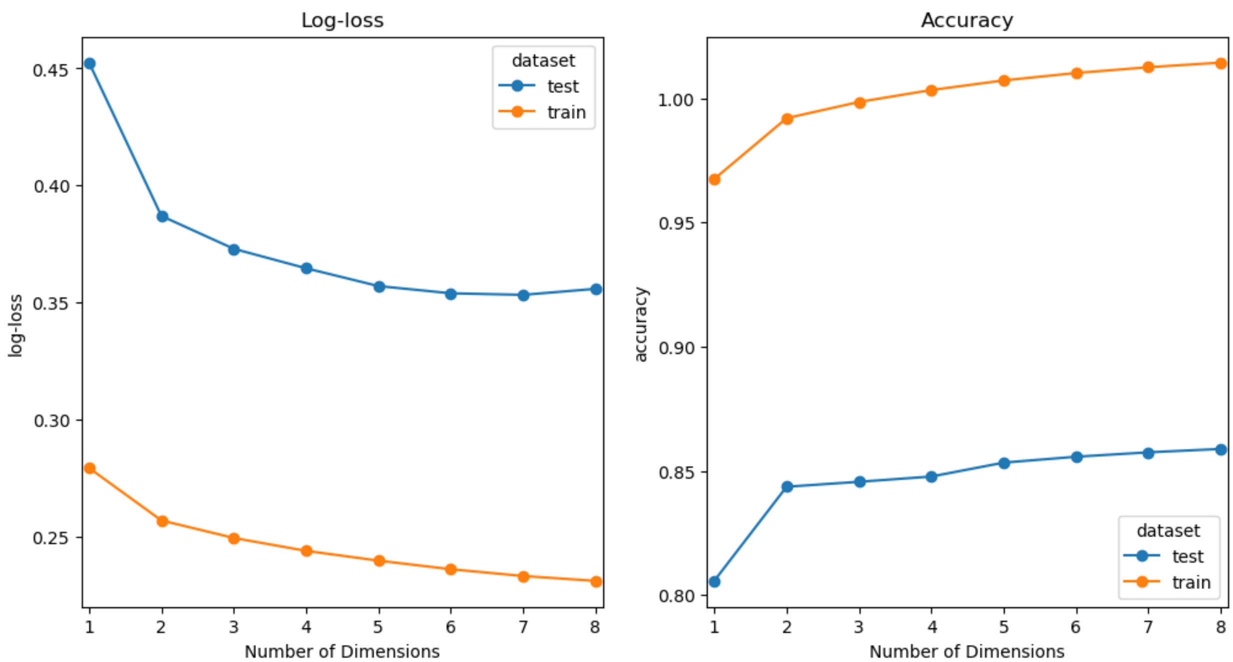

Figure 8: Metrics on Cosponsorship 93rd-115th Congresses

much noiser than votes. Since so many bills are introduced that do not proceed in a given session, there are likely many cases where a legislator would have preferred that a bill pass even if they did not have the opportunity to copsonsor it. Because of this, we add additional dropout layers on the bill parameters themselves. This limits the extent to which the model can overfit to noise in the dataset, although the results in 8 show that this is only somewhat successful as the training accuracy somewhat rapidly reaches 1.

While the above results could be construed as evidence, that there are fewer dimensions in cosponsoring decisions than there are in voting it seems equally plausible that the model is more easily fitting to noise in the cosponsorship decisions so rapidly that the true policy preference dimensions.

Including covariates

We also extend the model to include covariates. As an exmaple, it is plausible that a legislator might behave differently while in the party that controls the chamber. One way this might be observerd is that a legislator in the party in power is more likely to vote yes, on average, than someone in the minority party. This could even be the case when the match with their ideal point is very close, but due to party pressure a legislator could deviate from this to some extent. As such we estimate the models from the previous section including a variable indicating that a legislator is in the party in power so that the logistic layer now becomes:

$$prob(vote = yea) = \Phi\left(\beta\left[\exp\left(-\frac{1}{2}\sum_{k=1}^{s} w_k^2 d_{iyk}^2\right) - \exp\left(-\frac{1}{2}\sum_{k=1}^{s} w_k^2 d_{ink}^2\right)\right] + \gamma * pip_{ik}\right) \tag{4}$$

where γ represents the effect of being in the party in power. The results show that this factor is positive, as expected, but that the magnitude represents only a very small difference in vote probability. Note that the NN-Nominate framework does not easily lend itself to hypothesis testing. One way to achieve this would be a parametric bootstrap (ala Poole and Rosenthal) but this has not been attempted.[11]

Additionally, while a discussion of using a text representation of a bill, rather than a simple indicator value, is beyond the scope of this work this is a prime area of current research (for example see Kornilova, Argyle and Eidelman (2018)). These models work by building embedding representations of the bills themselves and then determining legislators ideal points relative to this more robust representation of a bill. While the accuracy of these models does not yet match that of the bill indicator approach, these models have the strong advantage of allowing out of sample prediction on new bill text. This means that we can determine how a legislator would vote on new bills, as they are introduced (for more we refer you to the **?**).

Conclusion

We have shown that ideal point estimation is a natural fit for a neural networks and that we can replicate existing parameterizations using this new system. However, we feel that this is only scratching the surface of their potential. We believe that extending this framework, both with ideas mentioned above and others, will yield new and important insights into the legislative process.

[11] One intriguing path forward is suggested by Mandt, Hoffman and Blei (2017) who show that under certain conditions stochastic gradient decent (as is used in neural networks) can proxy for Bayesian inference.

References

Alemán, Eduardo, Ernesto Calvo, Mark P Jones and Noah Kaplan. 2009. "Comparing Cosponsorship and Roll-Call Ideal Points." *Legislative Studies Quarterly* 34(1):87–116.

Brock, Andrew, Theodore Lim, James M. Ritchie and Nick Weston. 2016. "Neural Photo Editing with Introspective Adversarial Networks." *CoRR* abs/1609.07093.
URL: *http://arxiv.org/abs/1609.07093*

Carroll, Royce, Jeffrey B Lewis, James Lo, Keith T Poole and Howard Rosenthal. 2009. "Measuring bias and uncertainty in DW-NOMINATE ideal point estimates via the parametric bootstrap." *Political Analysis* 17(3):261–275.

Clinton, Joshua, Simon Jackman and Douglas Rivers. 2004. "The statistical analysis of roll call data." *American Political Science Review* 98(2):355–370.

Gerrish, Sean and David M Blei. 2011. Predicting legislative roll calls from text. In *Proceedings of ICML*.

Imai, Kosuke, James Lo and Jonathan Olmsted. 2016. "Fast estimation of ideal points with massive data." *American Political Science Review* 110(4):631–656.

Jackman, Simon. 2017. *pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory*. Sydney, New South Wales, Australia: United States Studies Centre, University of Sydney. R package version 1.5.2.
URL: *https://github.com/atahk/pscl/*

Kornilova, Anastassia, Daniel Argyle and Vlad Eidelman. 2018. Party Matters: Enhancing Legislative Embeddings with Author Attributes for Vote Prediction. Technical report.

Kraft, Peter, Hirsh Jain and Alexander M Rush. 2016. An Embedding Model for Predicting Roll-Call Votes. In *Proceedings of EMNLP*.

Mandt, Stephan, Matthew D Hoffman and David M Blei. 2017. "Stochastic gradient descent as approximate Bayesian inference." *The Journal of Machine Learning Research* 18(1):4873–4907.

Poole, Keith, Jeffrey Lewis, James Lo and Royce Carroll. 2011. "Scaling Roll Call Votes with wnominate in R." *Journal of Statistical Software* 42(14):1–21.
URL: *http://www.jstatsoft.org/v42/i14/*

Poole, Keith T and Howard Rosenthal. 1985. "A spatial model for legislative roll call analysis." *American Journal of Political Science* pp. 357–384.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15:1929–1958.
URL: *http://jmlr.org/papers/v15/srivastava14a.html*

Talbert, Jeffery C and Matthew Potoski. 2002. "Setting the legislative agenda: The dimensional structure of bill cosponsoring and floor voting." *Journal of Politics* 64(3):864–891.

Vorontsov, Eugene, Chiheb Trabelsi, Samuel Kadoury and Chris Pal. 2017. "On orthogonality and learning recurrent networks with long term dependencies." *CoRR* abs/1702.00071.
URL: *http://arxiv.org/abs/1702.00071*