

Legislator Arithmetic *

Daniel Argyle *FiscalNote*

See intro...

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. 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 networks, 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 different ideal point parameterizations such as DW-NOMINATE and the conventional Bayesian parameterization.

We demonstrate the algorithms 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.

Existing methods

[This is polmeth, y'all know this already]

This work is inspired by two, highly similar, lines of research in political science and computer science. Speaking generally¹, political scientists—from the days of (really old cite), through Poole

*The code for this method is available at the author's github repository.

¹A complete review of the literature in either field is beyond the scope of this work.]

and Rosenthal, and up to and including modern contributions such as (modern cites)—have focused primarily on ideal point estimation as a means to study the ideology space implied by the ideal points themselves. That these methods also predict votes is somewhat of an afterthought. On the other hand, computer science implementations largely focus on predicting legislator votes, without concern regarding the ideal points themselves.

Section that outlines the math and notation of WNOMINATE and the item response framework.

Combining the insights of these two fields, we wish to have the most predictive power without overfitting. We can then interpret the most predictive ideal points for insights.

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. that there is a clear tradeoff between explanatory power and ease of interpretation of ideal point models that should be explicit.² We implement ideal point models in a neural network framework because it's easily extensible and transparent. Our results suggest that the two dimensional model relied on sacrifices explanatory power in voting decisions.

Technical Details

While neural networks have become most strongly associated with artificial intelligence, they are, in essence, very flexible optimization systems.³ All that is required to estimate an ideal point model 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 WNOMINATE model as implemented in a neural network.

- Input layers: The inputs to this model consist of numeric legislator ids. The data is given as a batch of triples, $(legislator_id, bill_id, vote)$.
- Embedding layers: These layers take the numeric ids and convert them into continuous measures, for example from a specific legislator id to her ideal point. In this model there is one ideal point embedding for legislators, and two embeddings for bills, the `yes_point` and the `no_point`, that correspond to numeric values associated with WNOMINATE style estimation. The ideal point layer is restricted in two ways. First, the model has an orthogonality regularizer on the ideal points (cite). This penalizes correlation between ideal point dimensions and ensures that the resulting estimates are orthogonal. They also have a maximum norm constraint, which ensures that the norm of the ideal point vector for any legislator lies within the unit hypersphere.⁴
- Reshape layers: Because they are often used for text analysis, by default embedding layers assume that there is a sequence of ids 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.

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

³Indeed, there are several papers proving that neural networks can approximate any continuous estimator (cite)

⁴Note 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.

- JointWnomTerm layer: The embedded and reshaped data is fed into a custom neural network layer that implements the conventional WNOMINATE estimation (cite Poole and Rosenthal and the package). Specifically, the layer implements the following calculation

$$e^{-\frac{1}{2} \sum_{k=1}^s w_k^2 d_{iyk}^2} - e^{-\frac{1}{2} \sum_{k=1}^s w_k^2 d_{ink}^2}$$

where s 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.

- Dropout layer: Dropout regularization (cite original paper) 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 this specific iteration of the optimization algorithm. In practice, this has resulted in better model performance.
- Dense layer: The quantity obtained in the JointWnomLayer is then used as an input to a Dense layer, parameterised as a logistic regression (sigmoid activation and binary_crossentropy loss). The single model weight estimated here is denoted β in most NOMINATE models and represents [something that I'm having trouble recalling at the moment.]

The model begins with

We rely on several simple techniques from machine learning 1. Out of sample testing 2. Dropout regularization

Out of sample testing provides clear insight about the impact of a decision. Does it make sense to add another dimension? How much do dynamic ideal points help?

Dropout in two places: 1. Each dimension to ensure fitting even if first dimensions dominates. 2. On bill embeddings (to not overfit to a specific bill). Why not on legislators? The structure of the dynamic part of the model doesn't allow this easily.

Results

My method and WNOMINATE packages are similar

Let's do some things that weren't really feasible before

1. I implemented DW-NOMINATE as a neural network

- Why? Because I could! But also because it's a nice platform for this kind of optimization.
- It scales much better than existing implementations
- It's extensible in very interesting ways

2. All ideal point models are (a bit) overfit.

- At some point the algorithm starts to make marginal improvements to the parameters that don't improve out of sample performance

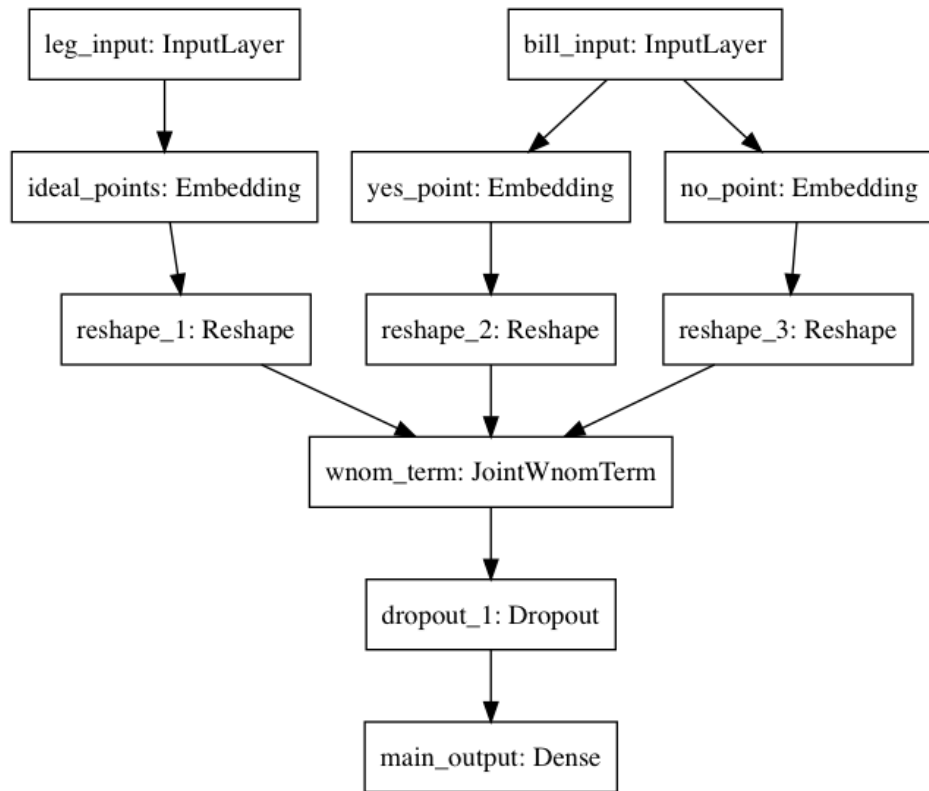


Figure 1: Base Model

- Out of sample performance matters much more than in sample (e.g. if we only cared about in sample we'd just add dimensions until we can predict it perfectly)
- Out of sample performance is a useful metric for evaluating modeling choices (adding another dimension, adding a time component, adding an external variable)

this is inline 42 and

```
py$answer
```

```
## [1] "42"
```

```
pass
```

1. How to implement in a neural network

- a. Discuss both parameterizations
- b. Show convergence plots
- c. Discuss out of sample fit as the metrics of importance

2. A Neural Network implementation works

- a. It estimates correctly on synthetic data (accuracy tables for both of these)
- b. It matches results from standard libraries (wnominate and pscl) on both synthetic and real data
 - i. Show both in sample and out of sample accuracy (will require writing a python wrapper to get predictions from the R results)

3. Results and applications of the neural network implementation

- a. What are the optimal number of dimensions? Plot these and discuss tradeoffs
- b. Applying the model to cosponsorship
- c. Polarization differences
 - i. Votes models over time (how to do in multiple dimensions?)
 - ii. Compare between cosponsorship and votes