

# 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 generally<sup>1</sup>, political scientists—from the days of (really old cite), through Poole

---

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

<sup>1</sup>A 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.

## Technical Details

### 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
  - 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"
```