Deep IV in Law: Appellate Decisions and Texts
Impact Sentencing in Trial Courts

Zhe Huang, Xinyue Zhang, Ruofan Wang, Daniel L. Chen

July 5, 2021

Abstract

Do U.S. Circuit Courts’ decisions on criminal appeals influence sentence lengths imposed by U.S. District Courts? We explore the use of high-dimensional instrumental variables to estimate this causal relationship. Using judge characteristics as instruments, we implement two-stage models on court sentencing data for the years 1991 through 2013. We find that Democratic, Jewish judges tend to favor criminal defendants, while Republican, Catholic judges tend to rule against them. We also find from our experiment that prosecutors backlash to Circuit Court rulings while District Court judges comply. Methodologically, we demonstrate the applicability of deep instrumental variables to legal data.

1 Introduction

This paper develops a method for conducting automated impact analyses of court precedent and applies it to criminal sentencing. This topic has received much attention due to the massive build-up of prisons in the United States criminal justice system. We apply methods from machine learning, natural language processing, and causal inference to measure the causal impact of criminal appeal decisions in Circuit Courts.

Legal theorists and historians have long debated the proper relationship between constitutional law and politics. While some have argued that judicial decision-making should be political (Schmitt 1969, 1985, 2005), most scholars have emphasized the importance of a separation from political interests. Debates over the political role of the judiciary have intensified in recent years. This paper assesses the impacts of ideological motivations of United States federal judges as reflected in their rulings and subsequent compliance by federal courts as evidence that this debate over judicial decision-making has consequences. We test the effects of legal
precedent in criminal justice on subsequent sentencing decisions of District Court judges and sentencing charges by federal prosecutors.

To conduct our analysis, we represent the text of judicial decisions as data. We then take these text features, along with metadata about the judges and case facts, to predict appeal court decisions (affirm/reverse) and district court sentencing decisions (length of sentence, in months). Using a high-dimensional instrumental variables approach, we measure the causal relations underlying these processes.

Our approach is based on Hartford et al (2017). The prediction problem is divided into a two-stage model. In the first stage, we fit models that learn to predict appeal decisions of circuit court as well as the vector representation of judge opinion text, where the instruments include characteristics of assigned judges. Intuitively, the Deep IV methods will be beneficial in predicting a high-dimensional embedding vector describing the text features of the written decision. In the second stage, we predict district court sentencing length decisions. These models use the first-stage predictions as inputs, so the resulting model parameters have a causal interpretation. We compare these Deep IV predictions to the non-causal Deep OLS predictions and the Deep Reduced Form predictions that use only the judge characteristics as regressors. We also report feature importance and OLS coefficients. The reduced form model is used to substantiate causality and aid in interpretability.

We find that an appeal case that affirms a lower-court crime decision (that is, a decision to be harsh) is followed by a statistically significant increase in sentencing percentile relative to sentencing guidelines in the lower-courts of that circuit. However, there is a statistically insignificant effect on sentence lengths. Sentence guidelines dictating the minimum and maximum is based on a formula using the prosecutor’s charge. We therefore interpret these results as being due to the interplay of prosecutors and judges, where prosecutors backlash to circuit rulings by issuing more lenient charges after a harsh ruling (or conversely, harsh charges after a lenient ruling), yet district judges are largely obeying the circuit rulings. This is consistent with the growing attention to the large role for discretion in decision-making by prosecutors.

2 Literature Review

There is an extensive research literature on the topic of judicial decision-making and sentencing. And it is clear that contextual factors related to political, judicial, and social environments affect prison sentences (Huang et al 1996). Regional variation in sentencing has been documented in a lot of research, both at the local (Fearn 2007) and district or circuit level (Kautt 2002). This paper examines the casual link between legal rulings on appeal decisions in circuit courts and the subsequent
sentencing decisions in the lower district courts within the circuit jurisdiction. We are unaware of any previous study of this causal question for sentencing, and more broadly, of how judicial writing style affects downstream outcomes.

In order to measure the causality, this paper considers sentencing length predictions to be influenced by latent covariates from various political, social, and economic factors. At the core of our methodology is the use of features generated from a naturally occurring random process in our prediction task. We exploit the fact that judges of each case are randomly assigned, and we take judge characteristics as an instrumental variable (Chen et al 2016).

Methodologically, previous work by Hartford et al (2017) indicate that when doing counterfactual predictions there is a benefit from a deep instrumental variable framework, which is a two-stage deep neural network (DNN) instrumental variables method. The Deep IV framework can outperform both traditional two-stage OLS and standard feed-forward network by significantly reducing counterfactual errors.

The field of counterfactual analysis has been developing fast, and has started gaining more attention from the machine learning community in recent years. Recent work from Lewis and Syrgkanis (2018) use Generative Adversarial Networks (GANs) and find that GANs has similar or better performance compared to both direct models and other forms of two-stage models. Egami et al (2017) also used a related method to measure treatment effects from text and showed applicability - however, in most of their papers, the model is tested on simulated data. In contrast, the focus of our paper is on real, complex data environment. Other papers that connect machine learning with estimating treatment effects in economics, law, and policy include Double ML (Chernozhukov et al., 2016), Causal Forest (Athey et al., 2019), Orthogonal Random Forest (Oprescu et al., 2019), etc.

3 Dataset Description

This paper construct the final dataset for analysis using four raw datasets. Here, we present brief descriptions for each of them.

3.1 Cleaned Circuit Court Case Data

First, we have raw text records of 253164 Circuit Court Opinions collected from 1891 to 2013, organized by year, case identification number, opinion type and author’s (judge’s) last name. It contains 82635 unique cases, 3288 unique judge names, and 14 unique opinion types. 75% of the cases are stated as being affirmed, and 25% are stated as being reversed.
3.2 Judge Biographical Characteristics

Second, we have demographic and background information for about 714 unique judges. The information contains a mixture of 186 numerical, text and categorical features, including the judges’ name, age, and party affiliation, as well as their education and career backgrounds.

3.3 District Courts Sentencing Data

Third, we have the dataset on District Court sentencing information. The feature we use here is the sentencing length, which is a numerical feature ranging from 0 to 999. 999 represents death sentence and hence will not be treated as numeric value. We use interquartile range (IQR) to detect outliers in the dataset, and thus consider data points with sentencing length greater than 152.5 as outliers. We eliminate those data from the analysis. The ones with missing values are also excluded from our analysis. The district courts sentencing data are later joined with circuit court data by using the U.S. state and the date of sentencing.

3.4 Circuit Cases Metadata

Fourth, we have a data set containing rich metadata for each circuit case, including the case-id, decision, date, three concurring judges, and case type. We use this table to filter out criminal cases that can be matched with opinion records to extract case and judge information.

4 Data Preprocessing

4.1 Feature Engineering

4.1.1 Demeaning Features

Many features in our data are potentially endogenous to court and time. For example, the number of Democrats in the court may be different each year and could have a confounding trend with outcomes. Since our data spread across 23 years, the changes over time might be significant. In addition, the cases are randomly assigned to judges conditional on circuit and year. We therefore demean instruments by circuit-year to reduce effects of confounding trends.

4.1.2 Target Calculation

We normalize the specified action string for the appeal decision to a binary variable, affirm or reverse. We group the 7 action categories using the rules in Table 1.
We are interested in measuring the effect of an appeal decision. Therefore, we set the target variable as the change in the average sentencing length before and after an appeal decision. To do this, we measure the sentencing length changes followed by a circuit court decision using the three months before and after the decision. We subtract the average sentencing length of 3 months before the decision from the average sentencing length of 3 months after the decision. This can be seen as a first-differenced outcome by case.

### 4.2 Representing Case Text as Data

Apart from the binary appeal action (affirm or reverse), we are also interested in whether the explanation for that action – the written opinion – might have a separate impact on sentencing decisions in the district court. To take account of this, we add text features to our treatment vector. The idea is that these embedded text features would represent some writing style characteristics that capture how judges reason toward sentencing decisions. We present two methods for representing textual features. First, we construct N-Gram frequencies and reduce dimensionality using Principal Component Analysis (PCA). Second, we use document embeddings.

#### 4.2.1 N-Gram model with PCA

Our first approach is to represent text using N-Grams. An N-Gram is a word sequence of length N. The N-Gram model represents a text document with a collection of N-Grams that appears in the text document.

There are multiple ways to featurize the N-Gram representation into a numeric vector. One of the simplest ways would be to denote the presence of an N-Gram

---

<table>
<thead>
<tr>
<th>Original Category</th>
<th>Grouped As</th>
</tr>
</thead>
<tbody>
<tr>
<td>stay, petition, or motion granted</td>
<td>Reversed</td>
</tr>
<tr>
<td>reversed (include reversed &amp; vacated)</td>
<td>Reversed</td>
</tr>
<tr>
<td>reversed and remanded (or just remanded)</td>
<td>Reversed</td>
</tr>
<tr>
<td>vacated &amp; remanded; set aside &amp; remanded; modified &amp; remanded</td>
<td>Reversed</td>
</tr>
<tr>
<td>vacated</td>
<td>Reversed</td>
</tr>
<tr>
<td>affirmed; or affirmed &amp; petition denied</td>
<td>Affirmed</td>
</tr>
<tr>
<td>petition denied or appeal dismissed</td>
<td>Affirmed</td>
</tr>
<tr>
<td>affirmed in part &amp; reversed in part; modified; affirmed &amp; modified</td>
<td>Dropped</td>
</tr>
<tr>
<td>affirmed in part, reversed in part, and remanded</td>
<td>Dropped</td>
</tr>
</tbody>
</table>
using boolean values of 0 and 1. Other simple ways include using the counts or frequencies of the N-Gram. However, these methods come with well-known issues, such as not capturing the importance of an N-Gram properly. Alternatively, we use Term Frequency-Inverse Document Frequency (TF-IDF) to score each N-Gram. The equations for calculating TF-IDF are shown in 1 - 3:

\[
TFIDF(t, d, D) = TF(t, d) \times IDF(t, D)
\]

\[
TF(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}
\]

\[
IDF(t, D) = \log\left(\frac{N}{\text{count}(d \in D, t \in d)}\right)
\]

where \(t, d, D\) denotes the N-Gram, the document and the corpus that contains \(N\) documents. \(TF(t, d)\) measures how frequent the N-Gram \(t\) occurs in current document \(d\), and \(IDF(t, D)\) measures how often the N-Gram appears across all document \(d\) in the corpus \(D\) with the intuition that if an N-Gram is common across many documents, then it is probably less informative about a particular document.

We convert the text documents into a TF-IDF matrix in python and then apply principal component analysis (PCA) to reduce dimensionality and keep only the largest 25 principle components.

We also experiment with simple counts as scores for each N-Gram. These counts featureize each document into a numeric vector. To compare the two methods, we use their resulting numeric representations of the document vectors to predict sentence length changes in an Ordinary Least Squares (OLS) regression model. We further experiment with using unigram (1-Gram) or bigram (2-Gram), in order to get better tradeoffs between representation power and computation cost. The result is shown in Table 2

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count (unigram)</td>
<td>1.762</td>
</tr>
<tr>
<td>Count (bigram)</td>
<td>1.138</td>
</tr>
<tr>
<td>TF-IDF (unigram)</td>
<td>1.088</td>
</tr>
<tr>
<td>TF-IDF (bigram)</td>
<td>0.917</td>
</tr>
</tbody>
</table>

Table 2: Comparison of TF-IDF and Count, OSL with PCA: Comparing using TF-IDF score versus simple count to represent a document vector in predicting sentence length change. Used PCA to reduce dimensionality of the document vector into 25 dimensions.

From the experiment, we saw that using TF-IDF is better than using simple counts. Furthermore, using bigrams gave better performance than using unigrams, which is intuitive. Further increase to 3-Grams substantially increases computation.
burden while the added benefit is slim. Therefore, for all the following experiments, we used bigrams with TF-IDF.

One of the limitations of this approach is the loss of information during dimensionality reduction. The information loss can be measured by the remaining explained variance of the selected principle components after PCA. We found that adding an additional principle component each time increased the explained variance by approximately 0.003, and even with 100 dimensions, the explained variance is just slightly above 10% of the total variance. This led us to seek a better method for representing text.

4.2.2 Document Embeddings

A better and more recent approach is to use document embeddings. Specifically, we used Doc2Vec model (Le and Mikolov 2014) to generate document embeddings for each case’s opinion text. We trained our document embedding model using text corpus containing all cases’ opinion text. We used GenSim Doc2Vec implementation with a context window of size 10, and generated a fixed size numeric vector of size 25 for each case’s opinion text. We show the scatter plot of these embedding vectors projected onto 2-D spaces in Figure 1

![Figure 1: 2-D projection of document vectors](image)

An important property of this model is that the geometric location of the embedding vector in high dimensional space encodes predictive information for the context-specific frequencies of words in the document. Intuitively, similar documents will be placed closer to each other in the embedding space. Le and Mikolov (2014) showed that the document vectors created with Doc2Vec outperformed other methods, including the popular Bag-of-Words model (BoW), for many natural language processing tasks. Figure 2 illustrates the idea of Doc2Vec model.
4.3 Normalization and Splitting Data

We aggregate the characteristics of the three judges in each circuit court case. We normalize all columns based on mean and standard deviation. After that, we randomly split the dataset into a training set, a validation set, and a test set. The final dataset has 7388 cases as rows. Columns contain 84 different features for 3 different judges, 25 extracted text features from the case's opinion, a binary column indicating appeal decision (affirm/reverse), and a target column indicating sentencing length changes.

A detailed description of the dataset and features is in Appendix 11.2. The descriptive statistics are based on values after demeaning and before normalization.

5 Empirical Model

The statistical approach is mainly based on the two-stage deep IV framework proposed by Hartford et al (2017), which is a high-dimensional generalization of the reduced form causal analysis approach described by Angrist (1996).

The deep IV framework assumes the structural form shown in equations 4 and 5 and defines the counterfactual prediction function as 6. The graphical model is illustrated in Figure 3.
\[ y = G(w, x) + e \quad (4) \]
\[ w = f(x, z, e) \quad (5) \]
\[ h(w, x) := G(w, x) + E[e|x] \quad (6) \]

where \( y, w, x, z, e \) are the target variable, the treatment variable, observed covariates, instruments and the error term that contains unobserved variables. The model further assumes \( E[e] = 0, E[e|x, w] \neq 0, E[we|x] \neq 0 \). With the use of instrumental variable \( z \) that satisfies the relevance assumption, exclusion assumption and unconfounded instrument assumption, counterfactual analysis we are interested in would be \( h(w_1, x) - h(w_0, x) \), where \( w_0 \) is the base treatment and \( w_1 \) is the target treatment, and \( h(w, x) \) is the solution to the inverse problem 8. Interested readers should refer to the Hartford et al (2017) for more details.

\[ E[y|x, z] = E[G(w, x)|x, z] + E[e|x] \]
\[ = \int h(w, x)dF[w|x, z] \quad (8) \]

In our study, the treatment variable (w) contains the appeal decision and accompanying opinion text features of the U.S. Circuit Court. Our outcome (y) is the sentencing length change (from 3 months before to 3 months after) in the corresponding district courts. The instrumental variable (z) is the randomly assigned circuit judge characteristic. The variable (x) contains possible covariates of the circuit case, such as detailed topic. The confounder (e) is correlated with the treatment variable (w) and the outcome (y) but not with the instruments (z).

To measure the effect of criminal appeal decisions in circuit courts on the changes in sentencing decisions of district courts, we are going to carry out three main prediction tasks.

First, what we call "Deep OLS," which involves training \( F(y|w) \). We train a model to predict the District Court sentencing length changes (y) using the appeal decision and opinion text features (w).

Second, what we call "Deep Reduced Form," which involves training \( F(y|z) \). We train a model to predict District Court sentencing length changes (y) from the judges characteristics (z).

Third, we have the "Deep IV" or "Deep 2SLS" approach. This is a machine learning implementation of the two-stage deep IV framework proposed by Hartford et al (2017). In the first stage, we will be training \( F(w|z) \), and predict the circuit court appeal decisions and opinions (w) using judge characteristics (z). There are 26 different target variables and we form a prediction \( \hat{w} \) and measure the \( R^2 \) for
each. In the second stage, we are predicting $y$ by learning a function $G(y|\hat{w})$. That is, we use the outcome of the circuit court appeal decisions and text features from first stage model to predict the sentencing length changes.

We compare the models on performance in prediction tasks and statistical tests. For predictability, we measure the out-of-sample mean squared error and $R^2$. The formula for computing $R^2$ is shown in equation 9, where $\hat{y}_i$ is the predicted sentence length change.

$$R^2 = 1 - \frac{\sum_i(y_i - \hat{y}_i)^2}{\sum_i(y_i - \bar{y})^2} \quad (9)$$

6 Results

6.1 Deep OLS

For the Deep OLS model, we use the extracted text features of circuit court case, which encodes the judge’s writing style, and circuit court decision (affirm/reverse) to directly predict district court sentencing length change $y$. We experimented with both N-Gram model with PCA approach and the document embedding approach. We also experimented with different algorithms to see the predictive power of the text features. Using only the text features, we compared different regression models. Among these, ensemble methods perform the best. The result of Decision Tree Regressor (J.R. Quinlan 1986), Support Vector Regressor (Cortes and Vapnik 1995), Gradient Boosting Regressor (Friedman 2001), and Random Forest Regressor (Liaw and Wiener 2002) are given in Table 3.

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>1.55</td>
<td>0.90</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>2.53</td>
<td>1.19</td>
</tr>
<tr>
<td>SVM</td>
<td>1.48</td>
<td>0.87</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>1.44</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Table 3: Model Comparison: predicting sentencing length changes with only text features from each case’s opinion text. Four popular machine learning models are used in the comparison. The performances are measured using mean squared error and mean absolute error.

We see that Gradient Boosting Regressor performs the best. We will later compare Gradient Boosting Regressor with a neural network.

Next, we included the circuit court appeal decision (reverse/affirm) as feature. We compared a two-layer neural network with Gradient Boosting Regressor. We implement the network in pytorch (Paszke et al., 2019). We applied dropout (Srivastava et al., 2014) and batch normalization (S.Ioffe and C. Szegedy 2015) to avoid
overfitting and facilitate training. We didn’t perform extensive hyper parameter searches on the neural network. We performed grid search on the Gradient Boosting Regressor to select the best hyper parameters within the search space using the validation set. The N-Gram model with PCA approach gave us a mean squared error of 0.82 and mean absolute error of 0.72 on the test set for Neural Network, and mean squared error of 0.62 and mean absolute error of 0.64 for Gradient Boosting Regressor. For this particular dataset and task, the Gradient Boosting Regressor performed slightly better than our Neural Network. We will discuss some possible reasons in Section 7.

For the document embedding approaches, the best performance is achieved using Gradient Boosting Regressor with Doc2Vec text representations (Le and Mikolov 2014); the mean square error is 0.65. We also experimented with another more recent document embedding method proposed by Arora et al (2017), and found that it did not achieve better performance than Doc2Vec for our dataset and task.

### 6.2 Deep Reduced Form

The Deep Reduced form analysis is to predict District Court sentencing length $y$ from the judges characteristics $z$ by training $F(y|z)$. We monitor the change of average sentencing length from 3 months before the circuit court decision to 3 months after the circuit court decision from the same circuit area demeaned by circuit-year.

We tried a range of models. We tried to fit Neural Network, Linear Regression, Ridge and Lasso Regression with RBF Kernel, Gradient Boosting Regressor and Random Forest Regressor. We used mean squared error to measure model performances.

Among these models, Random Forest Regressor performs the best. The hyper-parameter is chosen according to validation performance. The best mean squared error is 0.49. The scatter plot of true and predicted values is given in Figure 4. The instruments have clear predictive power, as the predicted value is increasing with the real value of the target. The $R^2$ of our prediction 0.094.
Next we plot the feature importance as reported from the random forest. We report this both with and without the demeaning step in Figure 5 and Figure 6 (description of the feature name can be seen in Appendix 11.2). We see that demeaning makes a big difference in terms of feature importance. Without demeaning, the most important features are mainly about the number of Republicans and Democrats, as well as the judges’ own party. After demeaning, the most important features include whether a judge is a Solicitor-General, the age of the judge, and the number of Republicans in the Senate at the year of appointment. This demonstrates the importance of potential confounders for the OLS estimates. Binscatter plots of two important demeaned features are also presented in Figure 7 and Figure 8.

Figure 4: Reduced Form Model Predicted Values Vs Real Target

![Figure 4: Reduced Form Model Predicted Values Vs Real Target](image)

Figure 5: Reduced form Feature Importance before Demeaning

![Figure 5: Reduced form Feature Importance before Demeaning](image)

Figure 6: Reduced Form Feature Importance after demeaning

![Figure 6: Reduced Form Feature Importance after demeaning](image)
The random forest feature importance ranking does not tell the direction of the effect of the predictors. To see the direction of the effects, we fit a linear regression separately for each of the top 10 important instruments in the reduced form. These coefficients are reported in Table 4. We can see that number of Republicans in Senate (at the time of appointment) increase sentence length and Older judges (those born in the 1910s) decrease sentence lengths.

<table>
<thead>
<tr>
<th>Features</th>
<th>Importance</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solicitor-General</td>
<td>0.0517</td>
<td>-0.1748</td>
</tr>
<tr>
<td>Born in 1910s</td>
<td>0.0416</td>
<td>-0.0611</td>
</tr>
<tr>
<td>Number of members of other political parties</td>
<td>0.0341</td>
<td>0.0489</td>
</tr>
<tr>
<td>Justice Department</td>
<td>0.0306</td>
<td>0.0346</td>
</tr>
<tr>
<td>Number of Republicans in the Senate</td>
<td>0.0282</td>
<td>0.0014</td>
</tr>
<tr>
<td>Age at time of commission</td>
<td>0.0281</td>
<td>-0.0011</td>
</tr>
<tr>
<td>Full-time law professor</td>
<td>0.0275</td>
<td>-0.0018</td>
</tr>
<tr>
<td>Deputy or assistant district/county/city attorney</td>
<td>0.0252</td>
<td>-0.0164</td>
</tr>
<tr>
<td>Born in 1940s</td>
<td>0.0244</td>
<td>0.0055</td>
</tr>
<tr>
<td>JD obtained in public school</td>
<td>0.0230</td>
<td>-0.0103</td>
</tr>
</tbody>
</table>

Table 4: Reduced Form Feature Importance and Regression Coefficient: Showing 10 important features according to feature importance by random forest regressor. Each of these feature is then fit individually in linear regressor to obtain the coefficient.

6.3 Deep2SLS

This section reports the results from a Deep2SLS approach for the impact of affirm/reverse on sentence lengths. We will predict circuit court appeal decision (af-
firm/reverse) and the text features (25 dimensions numeric values) in the first stage. We will then use the first-stage predictions to measure the treatment effect in the second stage.

### 6.3.1 First Stage

In the first stage, we predict circuit court appeal decision (affirm/reverse) and text features using judge characteristics. We have 84 features for each person. We sum the three values up based on each feature for the classification task.

For circuit court appeal decision, we experimented with several different classification models, including logistic regression (Cox 1958), gradient boosting (Friedman 2001), and random forest (Liaw and Wiener 2002). Area under the ROC Curve (AUC) was used as the evaluation metric for this task. The best AUC score is 0.86 on validation set, achieved by random forest classifier with $\text{maxdepth} = 6$ and $\text{numtrees} = 120$. The ROC curve and confusion matrix are reported in Figure 9 and Table 5. The F1 score of the categorical prediction is 0.18. The model’s other performance statistics are MSE=.145, RMS=.38, LogLoss=.46, and Gini=.204.

![Figure 9: ROC Curve: First Stage Prediction of Affirm/Reverse](image)

<table>
<thead>
<tr>
<th>Prediction:</th>
<th>Affirmed</th>
<th>Reversed</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Affirmed</td>
<td>470</td>
<td>419</td>
<td>0.471</td>
</tr>
<tr>
<td>Reversed</td>
<td>73</td>
<td>123</td>
<td>0.372</td>
</tr>
<tr>
<td>Total</td>
<td>543</td>
<td>542</td>
<td>0.453</td>
</tr>
</tbody>
</table>

Table 5: Confusion Matrix: First Stage Prediction of affirm/reverse. Showing the actual circuit court appeal decision by rows and the predicted circuit court appeal decision by columns.
The feature importance for predicting affirm/reverse decision are shown in Figure 10 and 11 (Description of the feature name can be seen in Appendix 11.2). The bar plots show the top 10 important features derived by Random Forest model. We can see that the feature ranking does not change nearly as much as it did in the reduced form. They are quite similar. Both the reduced form and first stage rely on random assignment. It could be coincidental that demeaning matters more in reduced form. The causal interpretation rests on demeaning. The first stage results being more similar with and without demeaning may be because judges have a much more direct effect on their own decisions (affirm/reverse) than they do on the decisions of the district court judges in their jurisdictions.

Using feature importance to guide our exploration, we further built a logistic regression model on several selected features of interest to see whether each of them is positively or negatively correlated with the target variable. These coefficients are reported in Table 6. We see that Democrat judges and Jewish judges are more likely to reverse lower-court decisions. These are pro-defendant, liberal decisions. In turn, Republicans and Catholics tend to affirm lower-court decisions. This means they are more conservative in this area.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Bar Association Rating</td>
<td>-0.064</td>
</tr>
<tr>
<td>Age at time of Commission</td>
<td>0.026</td>
</tr>
<tr>
<td>Catholic</td>
<td>0.122</td>
</tr>
<tr>
<td>Democrat</td>
<td>-0.049</td>
</tr>
<tr>
<td>Number of Democrats in the House</td>
<td>0.025</td>
</tr>
<tr>
<td>Number of Republicans in the House</td>
<td>0.005</td>
</tr>
<tr>
<td>Jewish</td>
<td>-0.036</td>
</tr>
<tr>
<td>Republican</td>
<td>0.049</td>
</tr>
<tr>
<td>Number of Democrats in the Senate</td>
<td>-0.053</td>
</tr>
<tr>
<td>Number of Republicans in the Senate</td>
<td>0.056</td>
</tr>
</tbody>
</table>

Table 6: First Stage Regression Coefficient: Showing 10 important features according to feature importance by random forest. Each of these feature is then fit individually in logistic regression to obtain the coefficient.

For text features, we used the document embedding of 25 dimensions we generated from Doc2Vec (Le and Mikolov 2014) as our target, and judge characteristics as input. Since the output is high-dimensional, we fit one regressor for every individual text feature dimension. In this scenario we still choose random forest as the regression model.

After fitting the models, we calculate the $R^2$ for every regressor on the test set. The $R^2$ for each dimension is reported in Table 7 and Figure 12. The mean $R^2$ is 0.03.

<table>
<thead>
<tr>
<th>Text Feature</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>R Squared</td>
<td>0.023</td>
<td>0.057</td>
<td>-0.003</td>
<td>0.068</td>
<td>0.031</td>
<td>-0.009</td>
<td>0.035</td>
<td>-0.002</td>
<td>-0.008</td>
<td>0.015</td>
</tr>
<tr>
<td>Text Feature</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>R Squared</td>
<td>-0.015</td>
<td>0.025</td>
<td>0.036</td>
<td>0.004</td>
<td>0.007</td>
<td>-0.009</td>
<td>0.059</td>
<td>0.007</td>
<td>0.057</td>
<td>0.008</td>
</tr>
<tr>
<td>Text Feature</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R Squared</td>
<td>0.055</td>
<td>0.013</td>
<td>0.016</td>
<td>0.005</td>
<td>0.034</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: First stage R Squared for Document embedding: Showing the $R^2$ for each dimension of the document embedding when fitting to the random forest regressor.
6.3.2 Second Stage

In the second stage, we used as input the output of first stage, and used a two-layers neural network to predict the district court sentencing length change. As in DeepOLS section, we applied dropout and batch normalization to the neural network to avoid overfitting and facilitate training. Using the same architecture as in DeepOLS section, we get a mean squared error of 0.6955, which is better than the result we get using neural networks for all types of text feature representations in the DeepOLS section. This demonstrates the applicability of deep instrumental variables to legal data. Although the neural network here still performs a little worse than Gradient Boosting Regressor in the DeepOLS section, we argue that this might be mainly caused by the small size of our data. A common wisdom is that neural networks usually outperform traditional machine learning algorithms as the dataset becomes larger. We hypothesize that with a larger dataset we can build deeper neural networks and achieve better performance.

To see the effect of circuit court appeal decisions (affirm/reverse) on district court sentencing, we used only the predicted (affirm/reverse) decision as predictor to predict sentencing length change $y$. Below, We also group the sentencing length changes by the predicted (affirm/reverse) as well as the actual (affirm/reverse) decision and compare the mean and variance. Table 8 and Table 9 show the details of various statistics.
By fitting the linear regression to predict sentencing length from predicted binary decision, setting Affirmed to 1 and Reversed to 0, we get the 2SLS coefficient of Affirm decision to be -0.0739, and it is a weak correlation. The interpretation is that affirming the lower court decision (that is, being harsh on criminal defendants) leads to a weak decrease in sentence lengths. The result seems counterintuitive.

Thus, we further investigated this issue by examining the effect of circuit court decision on district court sentencing length deviation from sentencing guidelines (more precisely, a percentile relative to the recommended sentence minimum and maximum). For a case $i$, we compute its percentile as follows:

$$\text{percentile}_i = \frac{\text{sentence-length}_i - \text{glmin}_i}{\text{glmax}_i - \text{glmin}_i} \quad (10)$$

To measure the impacts on the percentile relative to sentencing guidelines, we subtract the average percentile of 3 months before the decision from the average percentile of 3 months after the decision. Using this outcome variable, we find that the coefficient of Affirm decision to sentencing deviation is 0.00758, and standard error is 0.003. Since the recommended sentence length is based on the charges brought forward by the prosecutor, the statistically significant positive effects on percentile relative to sentencing guidelines and insignificant negative effects on actual sentence lengths is likely due to the interplay between prosecutors and judges.
When a circuit judge issues a harsh decision (affirming the lower court ruling), prosecutors backlash by issuing more lenient charges. In the meantime, district judges are largely obeying the circuit rulings, resulting in a positive effect.

7 Discussion

Our experiments suggest that the neural network benefits from using a two-stage model compared to direct DeepOLS approach, because after controlling for the same model architecture and feature representation method, the two-stage approach achieves lower mean squared error and higher $R^2$. This is evidence that the inclusion of instrumental variables is helping with measuring causal effects and reducing counterfactual errors during prediction.

After reducing latent variable variations using two-stage IV methods, we believe the prediction of sentencing length change from appeal decision and opinion text is causal. We interpret our results as suggesting that prosecutors are backlashing while judges are complying to circuit court decisions. Using the IV framework, we also believe this result is causal.

Our experimentation with models on different stages shows that ensemble methods generally provide best results in almost all model selections, including linear regression with kernel and neural networks. Using same data partition to compare, they generally have higher $R^2$ scores and lower Mean square Error. This is interesting as neural networks are generally more commonly used among high dimensional features. We suspect that this may be due to our relatively small dataset.

8 On the Practical Use of Deep IV for Law and Economics

Legal scholars and judges have long made and justified their arguments about laws and regulations with theories about the effects of these legal rules. The situation resembles the field of medicine a century ago: prior to the advent of clinical trials, there were only theories without rigorous causal evidence. A growing body of empirical research demonstrates that causal inference is possible when cases are randomly assigned to judges. Randomizing cases to judges with different decision-making tendencies generates the inference on the long-run causal impacts of those decisions. This raises the possibility of a law platform that has four parts: first, automatically identifying the nearest previous cases when a case appears; second, fast-decision classification of the prior cases’ directionalities; third, the use of document embeddings for low-dimensional representation of legal dicta and reasoning; fourth, the use
of judge embeddings based on the history of their writings and citations to predict their verdicts on cases. The latter can be used to support judges in estimating the potential impacts of their rulings on downstream economic outcomes.

Formally, given treatment variables (law) $w_i$, instrument variables (judge characteristics) $z$, target variables (outcomes) $y_i$, and covariates $x_i$, the Deep IV model involves two main steps. First, a model of choice $F$ is trained to predict the treatment variable $w$ using the instrument $z$ and covariates $x$. Then the predicted treatments $\hat{w}$, instead of the true treatments $w$, are used together with covariates $X$ to predict the target variable $y$ using another model of choice $G$. Deep IV model is a method for performing counterfactual analysis. The basic idea behind the model is to remove the effect from unobserved confounders using instrument variables so as to estimate the true treatment effect.

To use the Deep IV model in practice, one can implement the models themselves with the proper graphical model as discussed in Section 5. The key requirement is to identify proper instruments $z$ that only affect the target $y_i$ through the treatments $w_i$.

Alternatively, one can use the publicly available Deep IV implementation released as part of the EconML toolkit (Microsoft Research 2019). The EconML toolkit is a python package dedicated for estimating treatment effect via machine learning models. Beside Deep IV, the package also includes models like Double ML (Chernozhukov et al., 2016), Causal Forest (Athey et al., 2019), Orthogonal Random Forest (Oprescu et al., 2019), etc. In their Deep IV implementation, the model $F(w|z, x)$ is chosen to be a mixture density network (Bishop 2006).

The Deep IV module in the EconML toolkit allows the user to either predict the outcomes $y_i$ given treatment assignment $w_i$ and covariates $x_i$ or directly estimate the treatment effect, which is calculated as the difference in outcomes based on two treatment points (that is, the base treatment and the target treatment). We find that it can actually be extended to a suite of higher dimensional treatments and instruments. In our study, we have a high dimensional treatment vector. We can use the 5th percentile values of each dimension in the document embedding vector, together with the affirm decision as a base treatment. Additionally, we use the 95th percentile values of each dimension in the document embedding vector, together with the reverse decision as the target treatment.

8.1 Practical Considerations

There are several challenges when applying Deep IV to real world problems. Below, we discuss some common challenges and bring forth some suggestions on how to view and address them.
Sensitivity to hyperparameters and network architecture. Similar to many other machine learning methods, the Deep IV method is sensitive to hyperparameter settings. In the EconML Deep IV module, for example, the first stage network uses the mixture density model. The hyperparameter $K$, which controls the number of mixture components used, is usually an important hyperparameter that can have strong influence on final performance. Generally speaking, with larger $K$, the model has larger capacity and is able to fit more flexible models, but requires larger dataset. The network architecture (the number of layers and the number of neurons in each layer, for example) can also play an important role. Again, a larger model usually requires more data. Better computing may address this issue.

Difficulty in hyperparameter selection. In contrast to many machine learning models where the goal is to make good prediction on new data, counterfactual analyses ask the "What-if" question. This poses a unique challenge, since we do not have the ground truth. Without the ground truth, we cannot use the standard hyperparameter selection approach, where we use a validation set that are assumed to be from the same data distribution as the train set and test set, and use the model performance on the validation set to select hyperparameters. This problem is not unique to Deep IV and hence also concerns other counterfactual analysis methods. These methods are usually developed using synthetic data, where the researchers define the data generating process, and thus know the true treatment effect. A method is successful if it recovers the true treatment effect. However, when we apply the method to real world data, we do not know the true treatment effect.

A workaround would be to still use the standard validation set and to use the validation loss to select hyperparameters, hoping that the validation loss correctly reflect the actual performance in counterfactual analysis.

Randomness. Randomness in the estimation might be another practical concern. To be specific, even with same data split, hyperparameters and network architecture, there could still be (at times substantial) variation between different runs of the model. This is somewhat expected given that Deep IV usually utilizes neural networks as model of choice. Common sources of the randomness include random initialization of the network weights, randomness caused by optimization algorithms like stochastic gradient descent, and randomness caused by the use of regularization methods like dropout (Srivastava et al., 2014).

One way to address this issue is to set a random seed at the beginning. Another common practice within the machine learning community would be to average the results across multiple runs to get a more reliable estimate of the true treatment effect.

However, depending on the actual problem and dataset, if the variation is too
large, this might also indicate issues such as not having enough data, not choosing the suitable architecture for the problem or not satisfying some of Deep IV’s assumptions.

Overall, we are excited about fast development in counterfactual analysis with machine learning. However, we would like to emphasize that these models should be used with caution and in conjunction with theories in economics and law, while the results should be interpreted from the context of policy and theory.

9 Limitations

Most of our predictions have a low $R^2$, which generally indicates a poor fit for prediction purposes. Although our data on all cases is substantial, we still argue that the aforementioned problem might be largely caused by the small size of the available data. The number of criminal cases that have both opinion data and all judges’ characteristics is not very large. Furthermore, the district sentencing data only ranges from 1991 to 2013, which is also limiting our selection. The final dataset we use for modeling only contains 7388 data points, which could be too small given this particularly challenging problem. Accordingly, we think the predictability of models is hugely affected by this, in part explaining our $R^2$ and F1 score results. Also, if we look at the distribution of our target variable (difference in sentencing length), we see that majority of the data gathers around value 0.2. The data distribution makes the horizontal line at mean value a hard benchmark to beat.

When predicting the circuit court appeal decision, our target variable (appeal decision) is imbalanced and our F1 score unsatisfying. To deal with this issue, we could try to down-sample or up-sample our data so as to make the model more robust. In our case, however, this might make our dataset even smaller.

When representing data, we also didn’t substantially explore all possible dimension space. Trying out different size of text feature representations may yields better modeling results.

Another limitation is that, due to the availability of data, we did not consider latent covariates in the two-stage model, which Hartford et al (2017) included. We think including covariates in the models will also help with increasing the overall predictability of models.

10 Potential Future Work

This study provides experiments concerning causal analyses of criminal sentencing. Future work may include expanding data to a larger time range, adding historical features of judges’ writing style, including covariates in the two-stage model, and
further fine-tuning of all models. We hope this work offers some insights and results for using two-stage deep IV models in causal investigations of law and judges’ decision-making.
11 Appendix

11.1 Distribution of predicted sentence length change

As an additional comparison, we plotted the distribution of the predicted sentence length change $\hat{y}$ for the main specifications using a neural network. These are reported in Figures 13 through 15. We can see that the distribution is quite different across the specifications. It shows that there is some omitted variable bias in the OLS specification, which has been corrected in the 2SLS specification.

Figure 13: Distribution of predicted target using doc2vec DeepOLS

Figure 14: Distribution of predicted target using Deep Reduced Form

Figure 15: Distribution of predicted target using doc2vec Deep2sls
11.2 Detailed description of the dataset and features

Table 10: Judge characteristics: We aggregate the characteristics of the three judges in each circuit court case, and demean by circuit-year
References


