Course Recap

Elliott Ash

Max Planck Summer School 2017

Course Recap

- These slides provide a recap of what we have done in this course:
  - 01 - Introduction to Corpora
  - 02 - Turning a Corpus into Features
  - 03 - Topic Models
  - 04 - Word Embeddings
  - 05 - Similarity and Clustering
  - 06 - Regression
  - 07 - Classification
  - 08 - Research Design

- We started with a corpus – a set of raw documents D.
  - We used a robot browser to download data from the web
  - We cleaned out HTML mark-up and other noise material
  - We discussed how to organize a corpus along with associated metadata.

01 - Introduction to Corpora

- We started with a corpus – a set of raw documents D.
  - We used a robot browser to download data from the web
  - We cleaned out HTML mark-up and other noise material
  - We discussed how to organize a corpus along with associated metadata.

01 - Introduction to Corpora

- We started with a corpus – a set of raw documents D.
  - We used a robot browser to download data from the web
  - We cleaned out HTML mark-up and other noise material
  - We discussed how to organize a corpus along with associated metadata.

- We started with a corpus – a set of raw documents D.
  - We used a robot browser to download data from the web
  - We cleaned out HTML mark-up and other noise material
  - We discussed how to organize a corpus along with associated metadata.

02 - From Corpus to Features

- We learned the major steps to tranform the corpus D to a matrix X:
  - split into sentences and words
  - exclude noise features: capitalization, punctuation, stopwords, and stems
  - compute basic corpus statistics for number of documents, sentences, words written, sentence length, word length, etc.

02 - From Corpus to Features

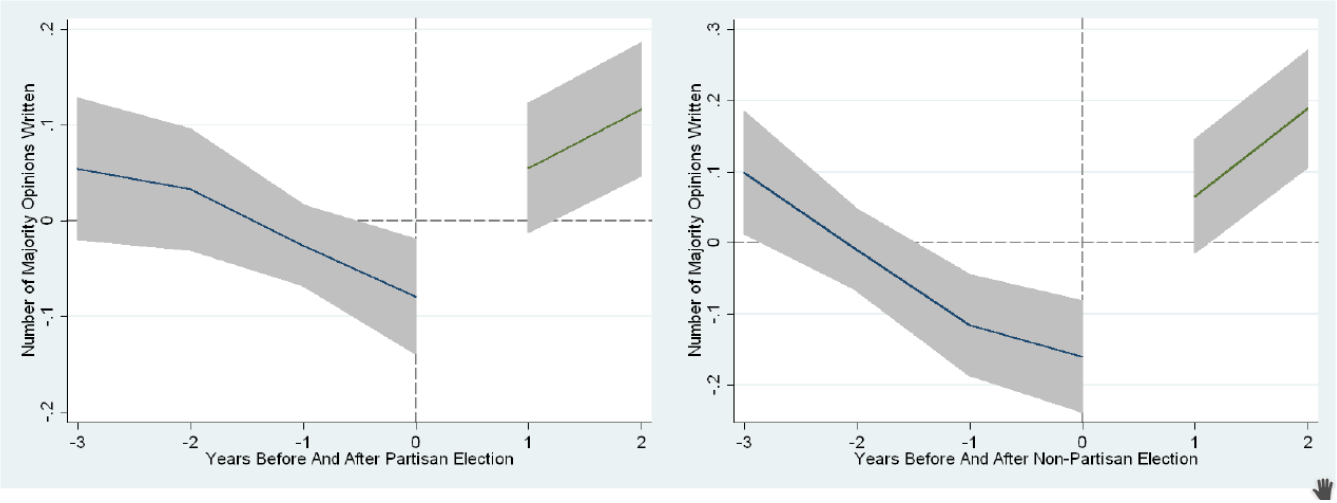
- We learned the major steps to tranform the corpus D to a matrix X:
  - split into sentences and words
  - exclude noise features: capitalization, punctuation, stopwords, and stems
  - compute basic corpus statistics for number of documents, sentences, words written, sentence length, word length, etc.

02 - From Corpus to Features

- We learned the major steps to tranform the corpus D to a matrix X:
  - split into sentences and words
  - exclude noise features: capitalization, punctuation, stopwords, and stems
  - compute basic corpus statistics for number of documents, sentences, words written, sentence length, word length, etc.

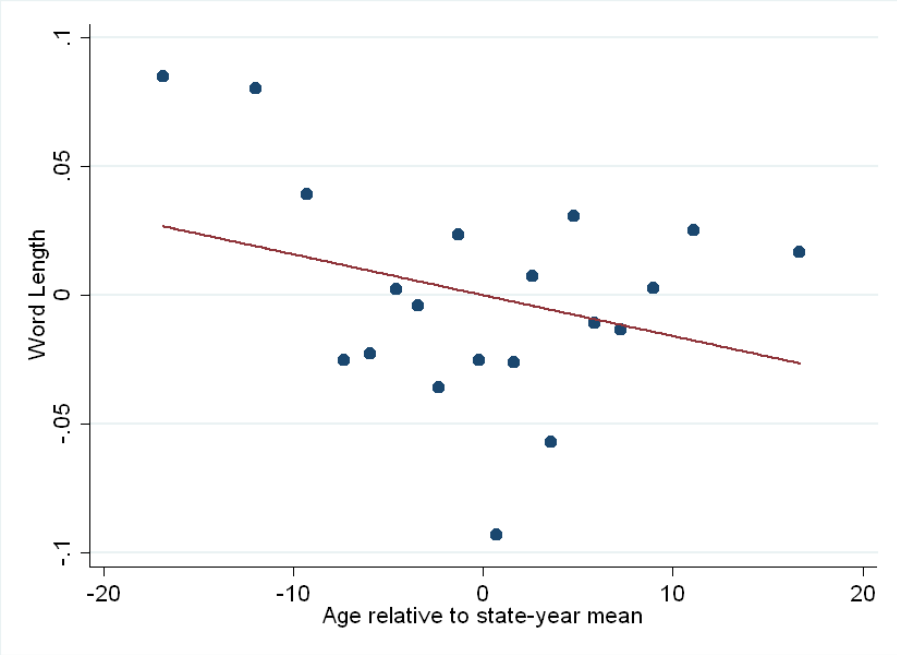
Elections Reduce Number of Opinions Written

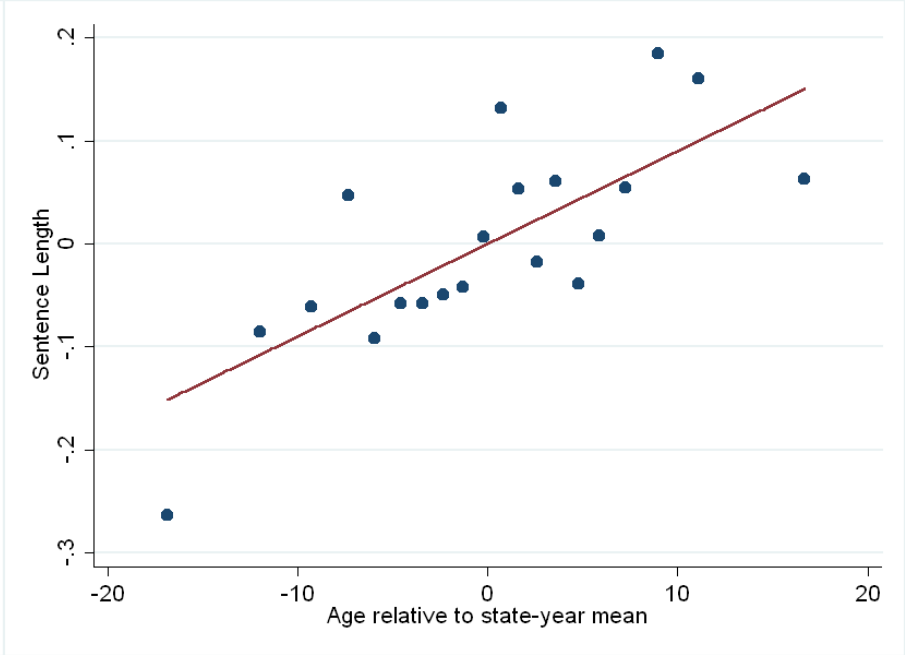
- Left panel: Partisan Elections, Right panel: Non-Partisan Elections



Fractional-polynomial prediction plots with y = outcomes and x = years before and after election year; outcomes residualized on judge and year fixed effects and standardized by judge; gray bars give 95% confidence intervals. Source: Ash and MacLeod (2016).

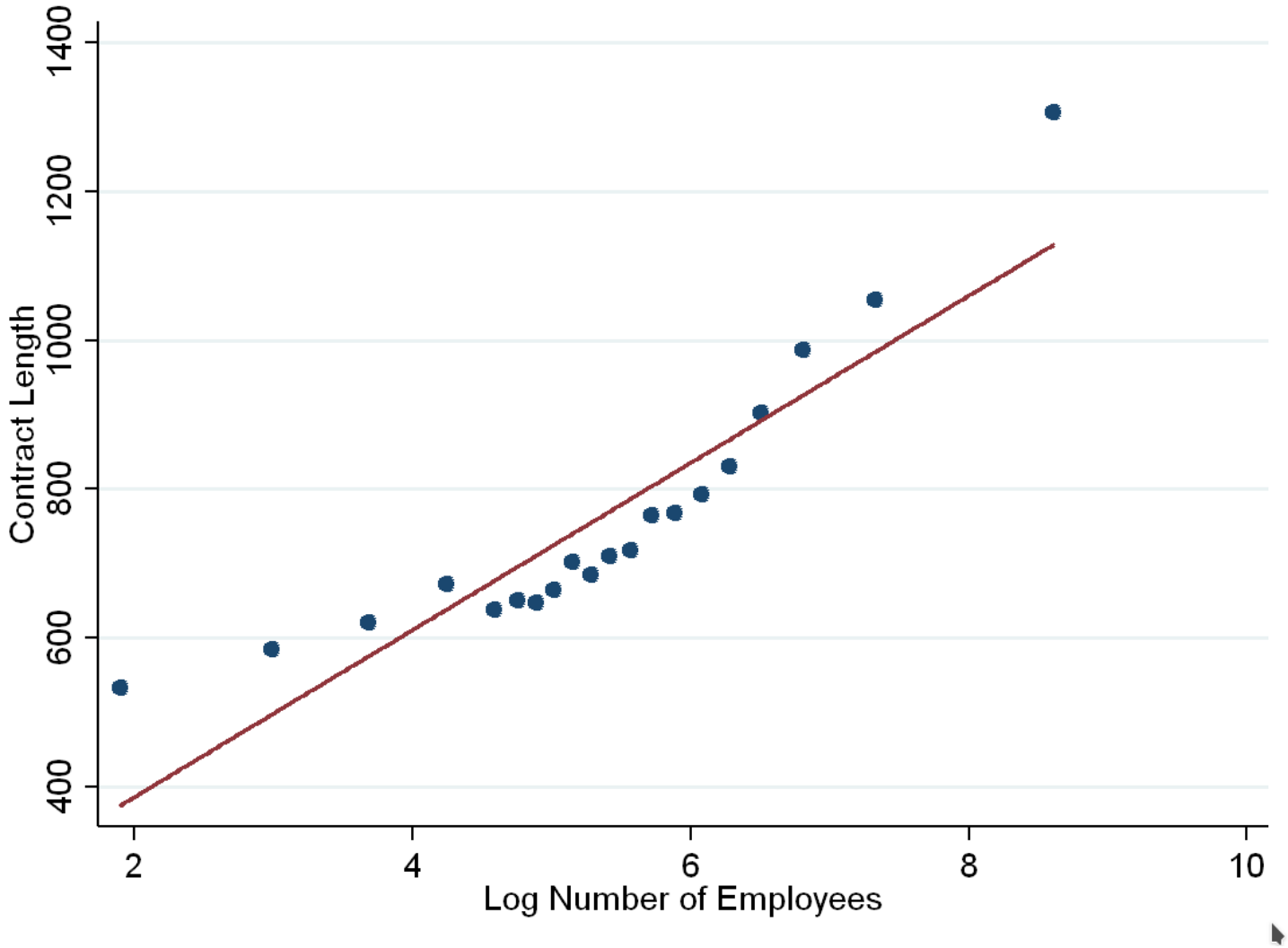
Judge Writing Style by Age





- Older judges use shorter words but longer sentences (Ash and MacLeod 2017).

Union Contract Length vs. Log Number of Employees



- Source: Ash, MacLeod, and Naidu (2017)

02 - From Corpus to Features - 2

- We tagged parts of speech: verb (VB), noun (NN), pronoun (PR+DT), adjective (JJ), adverb (RB), preposition (IN), conjunction (CC), and interjection (UH).
- We looked at dictionary-based methods, for example checking for the number of positive versus negative words in documents to examine sentiment.
- We constructed N-grams (short phrases) and discussed how to select the most-informative N-gram set (minimum frequency thresholds, point-wise mutual information, parts-of-speech sequences).
- We used a syntactic dependency parser to extract relations (obligations and entitlements) among agents in labor union contracts.

02 - From Corpus to Features - 2

- We tagged parts of speech: verb (VB), noun (NN), pronoun (PR+DT), adjective (JJ), adverb (RB), preposition (IN), conjunction (CC), and interjection (UH).
- We looked at dictionary-based methods, for example checking for the number of positive versus negative words in documents to examine sentiment.
- We constructed N-grams (short phrases) and discussed how to select the most-informative N-gram set (minimum frequency thresholds, point-wise mutual information, parts-of-speech sequences).
- We used a syntactic dependency parser to extract relations (obligations and entitlements) among agents in labor union contracts.

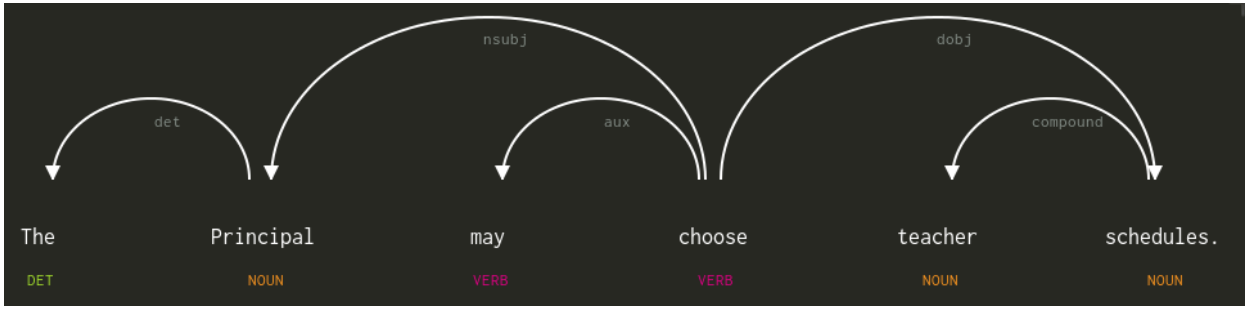
02 - From Corpus to Features - 2

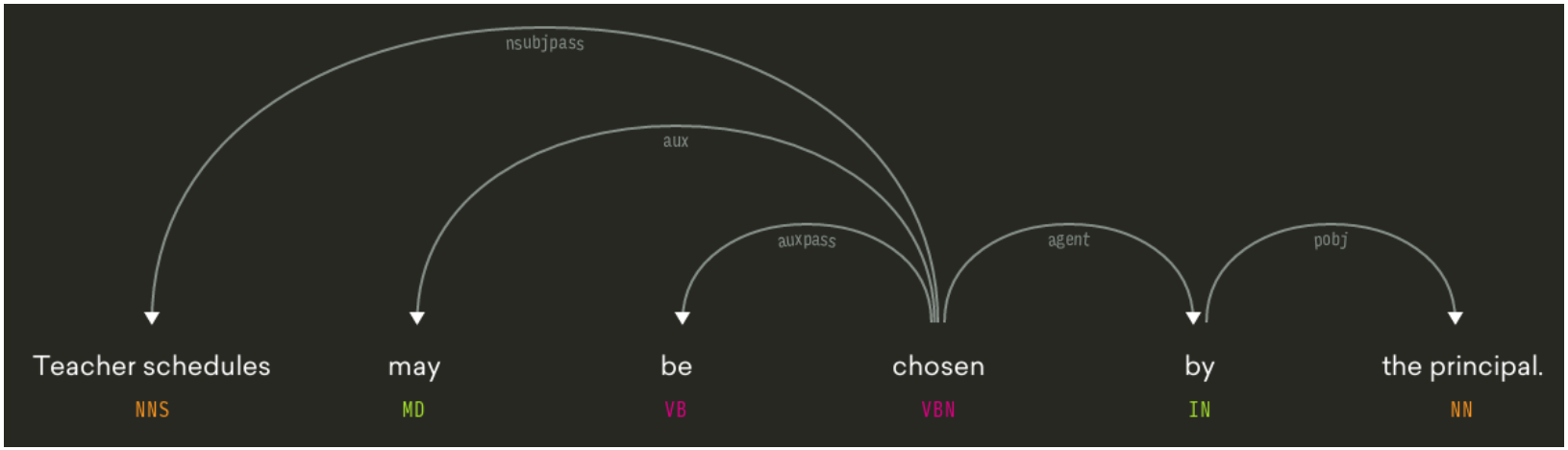
- We tagged parts of speech: verb (VB), noun (NN), pronoun (PR+DT), adjective (JJ), adverb (RB), preposition (IN), conjunction (CC), and interjection (UH).
- We looked at dictionary-based methods, for example checking for the number of positive versus negative words in documents to examine sentiment.
- We constructed N-grams (short phrases) and discussed how to select the most-informative N-gram set (minimum frequency thresholds, point-wise mutual information, parts-of-speech sequences).
- We used a syntactic dependency parser to extract relations (obligations and entitlements) among agents in labor union contracts.

02 - From Corpus to Features - 2

- We tagged parts of speech: verb (VB), noun (NN), pronoun (PR+DT), adjective (JJ), adverb (RB), preposition (IN), conjunction (CC), and interjection (UH).
- We looked at dictionary-based methods, for example checking for the number of positive versus negative words in documents to examine sentiment.
- We constructed N-grams (short phrases) and discussed how to select the most-informative N-gram set (minimum frequency thresholds, point-wise mutual information, parts-of-speech sequences).
- We used a syntactic dependency parser to extract relations (obligations and entitlements) among agents in labor union contracts.

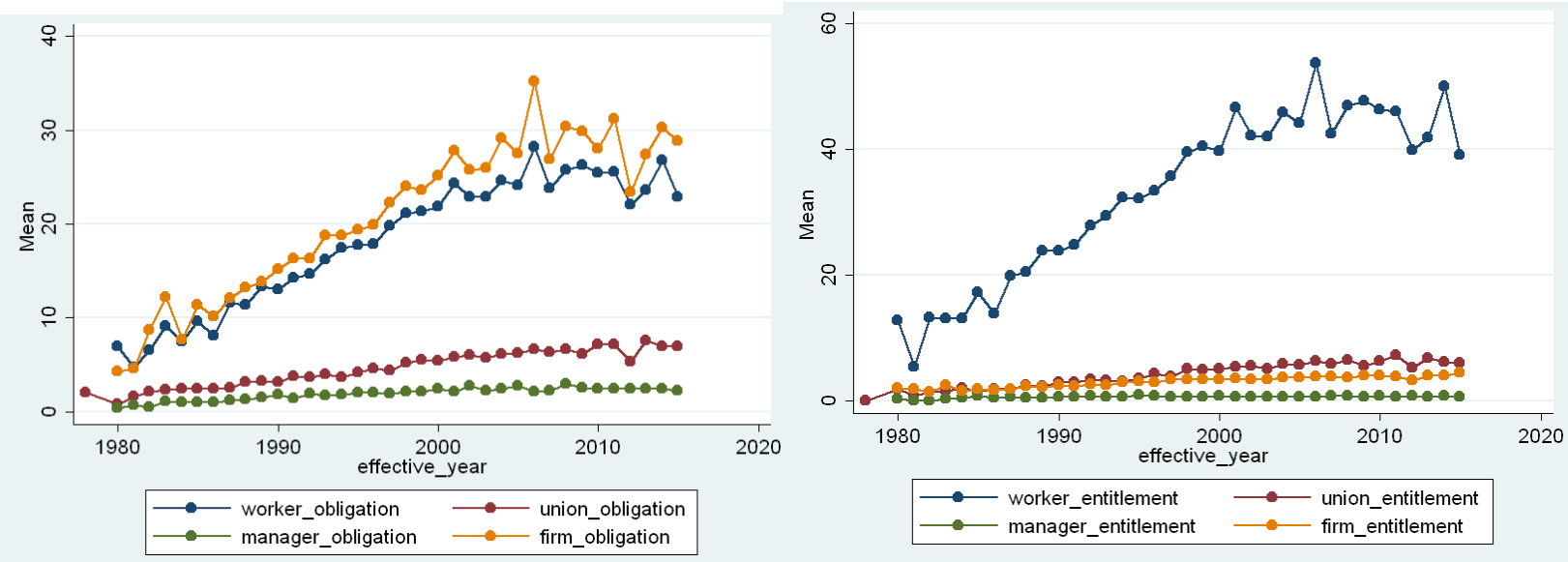
Syntactic Parsers





- The parser transforms sentences into parse trees, which represent the relations between words in a recursive hierarchical structure.

Obligations and Entitlements over Time, By Agent



- Workers and firms have gotten more obligations over time.
- Only workers have gotten more entitlements over time.
- Source: Ash, MacLeod, and Naidu (2017).

03 - Topic Models

- We used topic models as our key tool for unsupervised learning – letting the computer summarize and describe our high-dimensional data set.
- LDA reads the corpus and learns a set of topics (distributions over co-occurring words) and assigna them to documents.

03 - Topic Models

- We used topic models as our key tool for unsupervised learning – letting the computer summarize and describe our high-dimensional data set.
- LDA reads the corpus and learns a set of topics (distributions over co-occurring words) and assigna them to documents.

LDA Topics in Union Contract Sections

- 1 -- "**Sick Leave**" -- period month sick leave six probationary credit three complete employment twelve absent completion accumulate date exceed consecutive professional
- 2 -- "**Parental Leave**" -- leave absence pay request date grant prior week parental commencement pregnancy write maternity duty witness advance approve notice
- 4 -- "**Payroll**" -- change due result deduction amount status deduct monthly payroll reduction affect cheque technological fee employment orientation statement
- 5 -- "**Bargaining Unit**" -- unit bargaining person appointment appoint employ outside activity membership represent agent terminal sole select exercise ontario bargain behalf
- 7 -- "**Overtime**" -- hour shift work schedule overtime period call rest meal half minute start end break duty sunday weekend saturday two friday
- 8 -- "**Grievances**" -- grievance party procedure arbitration writing decision write step matter arbitrator committee complaint submit final dispute request name process
- 9 -- "**Job Training**" -- requirement operation training require equipment individual meet service responsibility provide program area manner performance" business duty operational
- 10 -- "**Vacation Leave**" -- year vacation service pay date employment week continuous effective two annual entitlement percent january salary earn termination period follow

04 - Word Embeddings

- Word embeddings provide a method to represents words as vectors, with this type of intriguing result:

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

- "You shall know a word by the company it keeps"

  - J.R. Firth, Papers in Linguistics,1957

- "He filled the **wampimuk**, passed it around and we all drunk some."

- Caliskan et al (2017) provide a "Word Embedding Association Test" and demonstrate the inherent biases in our language:
  - e.g., "man" relatively close to "programmer/engineer" and "woman" relatively close to "nurse/librarian"

04 - Word Embeddings

- Word embeddings provide a method to represents words as vectors, with this type of intriguing result:

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

- "You shall know a word by the company it keeps"

  - J.R. Firth, Papers in Linguistics,1957

- "He filled the **wampimuk**, passed it around and we all drunk some."
- Caliskan et al (2017) provide a "Word Embedding Association Test" and demonstrate the inherent biases in our language:
  - e.g., "man" relatively close to "programmer/engineer" and "woman" relatively close to "nurse/librarian"

04 - Word Embeddings

- Word embeddings provide a method to represents words as vectors, with this type of intriguing result:

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

- "You shall know a word by the company it keeps"

  - J.R. Firth, Papers in Linguistics,1957

- "He filled the **wampimuk**, passed it around and we all drunk some."

- Caliskan et al (2017) provide a "Word Embedding Association Test" and demonstrate the inherent biases in our language:
  - e.g., "man" relatively close to "programmer/engineer" and "woman" relatively close to "nurse/librarian"

04 - Word Embeddings

- Word embeddings provide a method to represents words as vectors, with this type of intriguing result:

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

- "You shall know a word by the company it keeps"

  - J.R. Firth, Papers in Linguistics,1957

- "He filled the **wampimuk**, passed it around and we all drunk some."

- Caliskan et al (2017) provide a "Word Embedding Association Test" and demonstrate the inherent biases in our language:
  - e.g., "man" relatively close to "programmer/engineer" and "woman" relatively close to "nurse/librarian"

Rudolph and Blei: Dynamic Word Embeddings

computer	
1858	**1986**
computer	computer
draftsman	software
draftsmen	computers
copyist	copyright
photographer	technological
computers	innovation
copyists	mechanical
janitor	hardware
accountant	technologies
bookkeeper	vehicles

bush	
1858	**1990**
bush	bush
barberry	cheney
rust	nonsense
bushes	nixon
borer	reagan
eradication	george
grasshoppers	headed
cancer	criticized
tick	clinton
eradicate	blindness

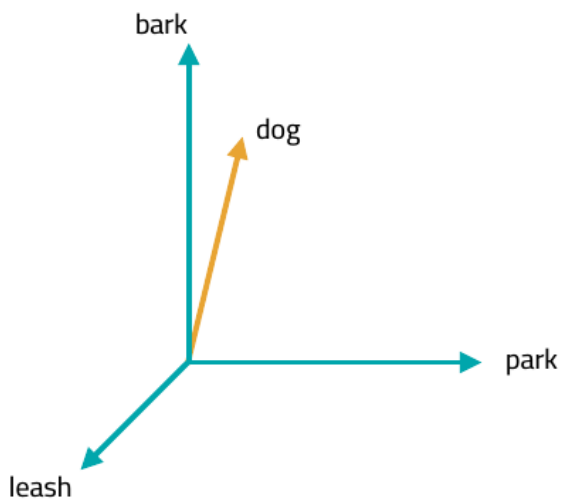
05 - Similarity and Clustering

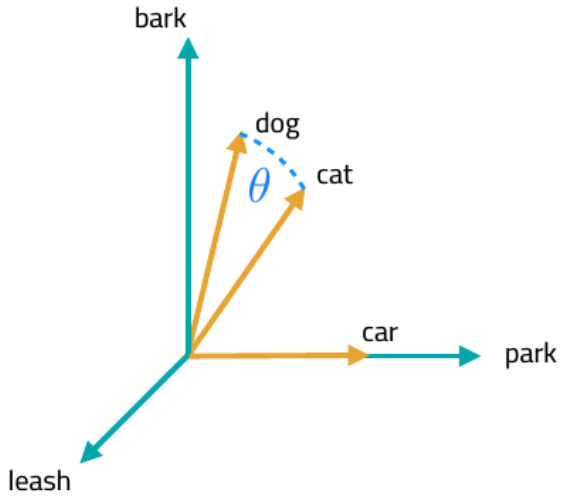
- We used cosine similarity and word mover similarity to compare documents and word vectors.
- We used k-means clustering to provide a discrete-groups alternative to topics.

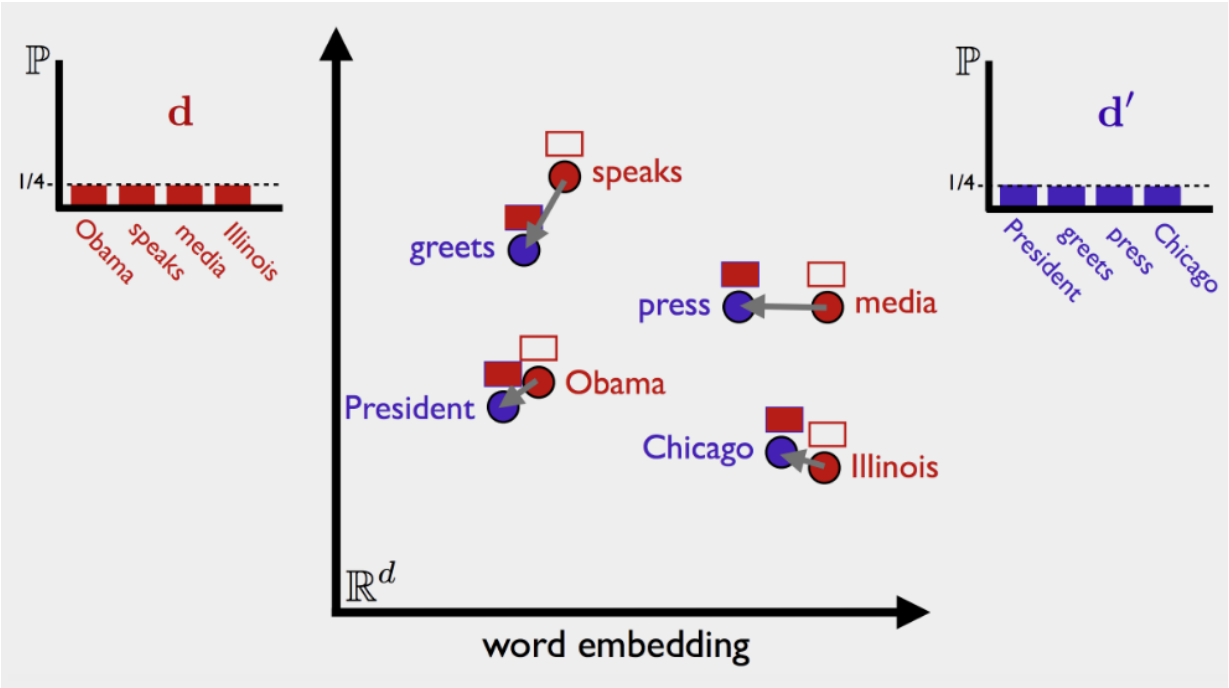
05 - Similarity and Clustering

- We used cosine similarity and word mover similarity to compare documents and word vectors.
- We used k-means clustering to provide a discrete-groups alternative to topics.

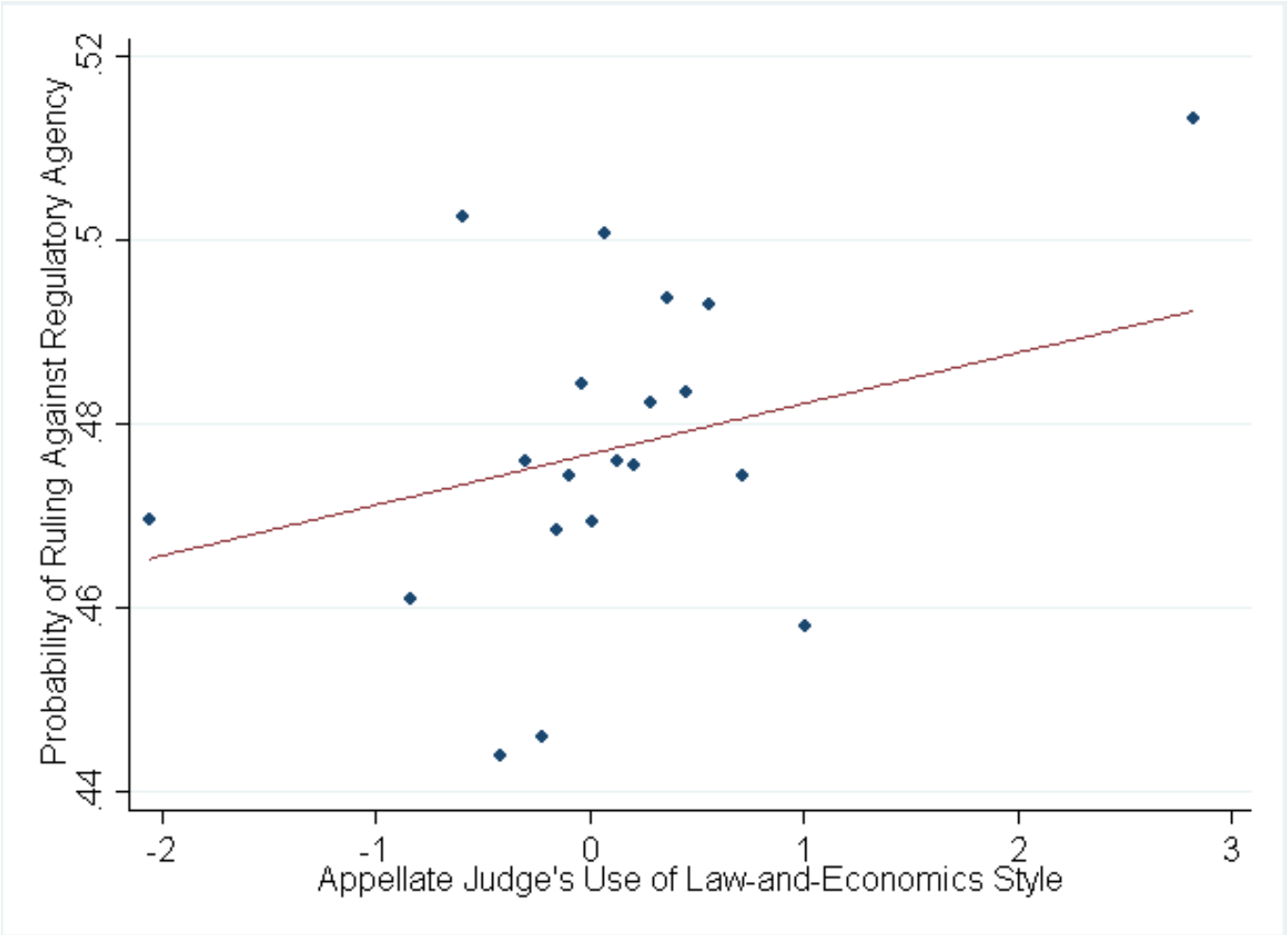
Measure distance between documents and words





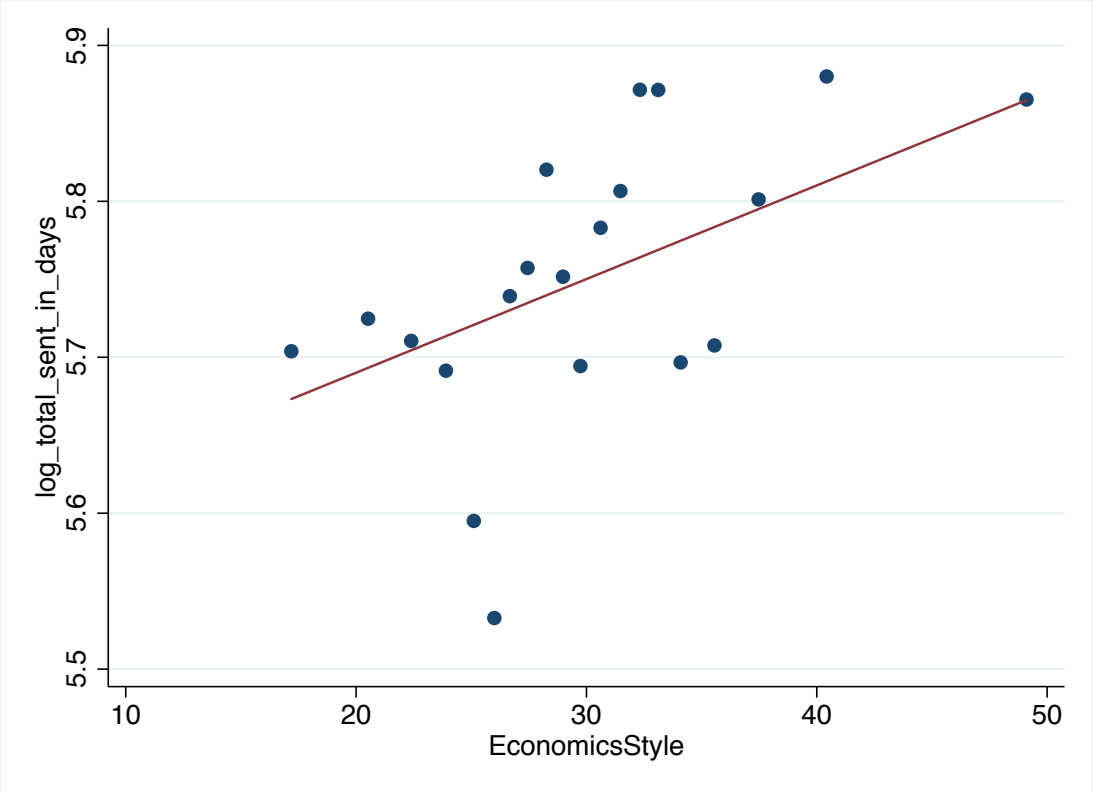


Judge Econ Language and Conservative Regulatory Decisions



- Source: Ash, Chen, and Naidu (2017)

Judge Economics Language and Sentence Length at Trial



- Source: Ash, Chen, and Naidu (2017)

06 - Regression

- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

06 - Regression

- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

06 - Regression

- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

06 - Regression

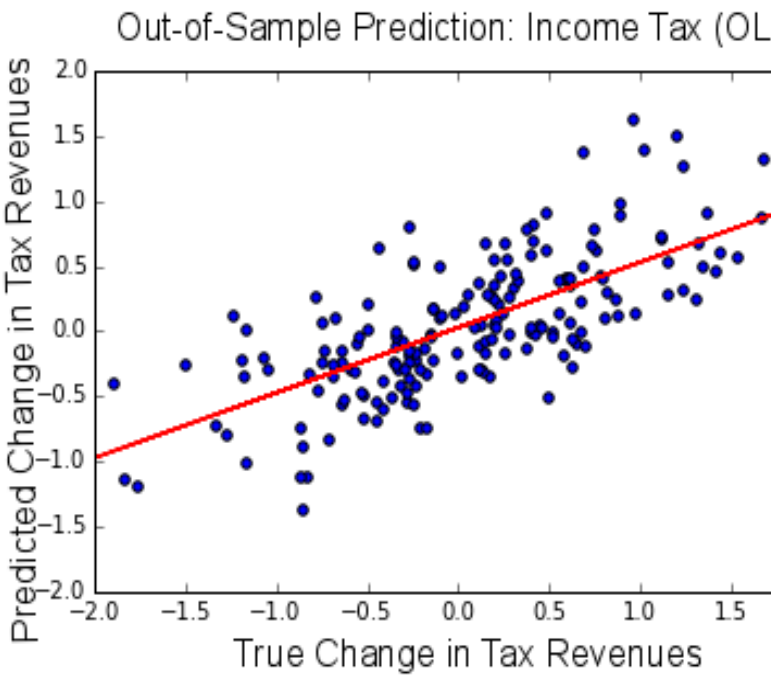
- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

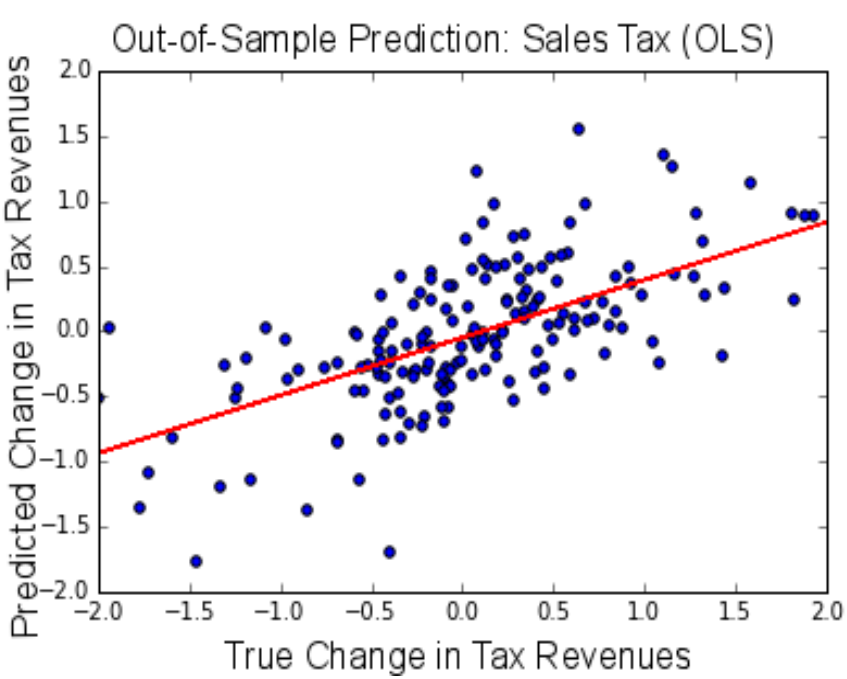
- Finally we used all of these features for prediction tasks – learning the function $f(\cdot)$ such that $Y = f(X)$.
  - Univariate OLS regression for feature ranking and filtering
  - Train/test splits for out-of-sample validation
  - Learning Models:
    - Principal component regression
    - Partial Least Squares
    - Elastic Net
    - Random forests
    - XGBoost

Out-of-sample PLS predictions of tax revenue changes

Income Tax



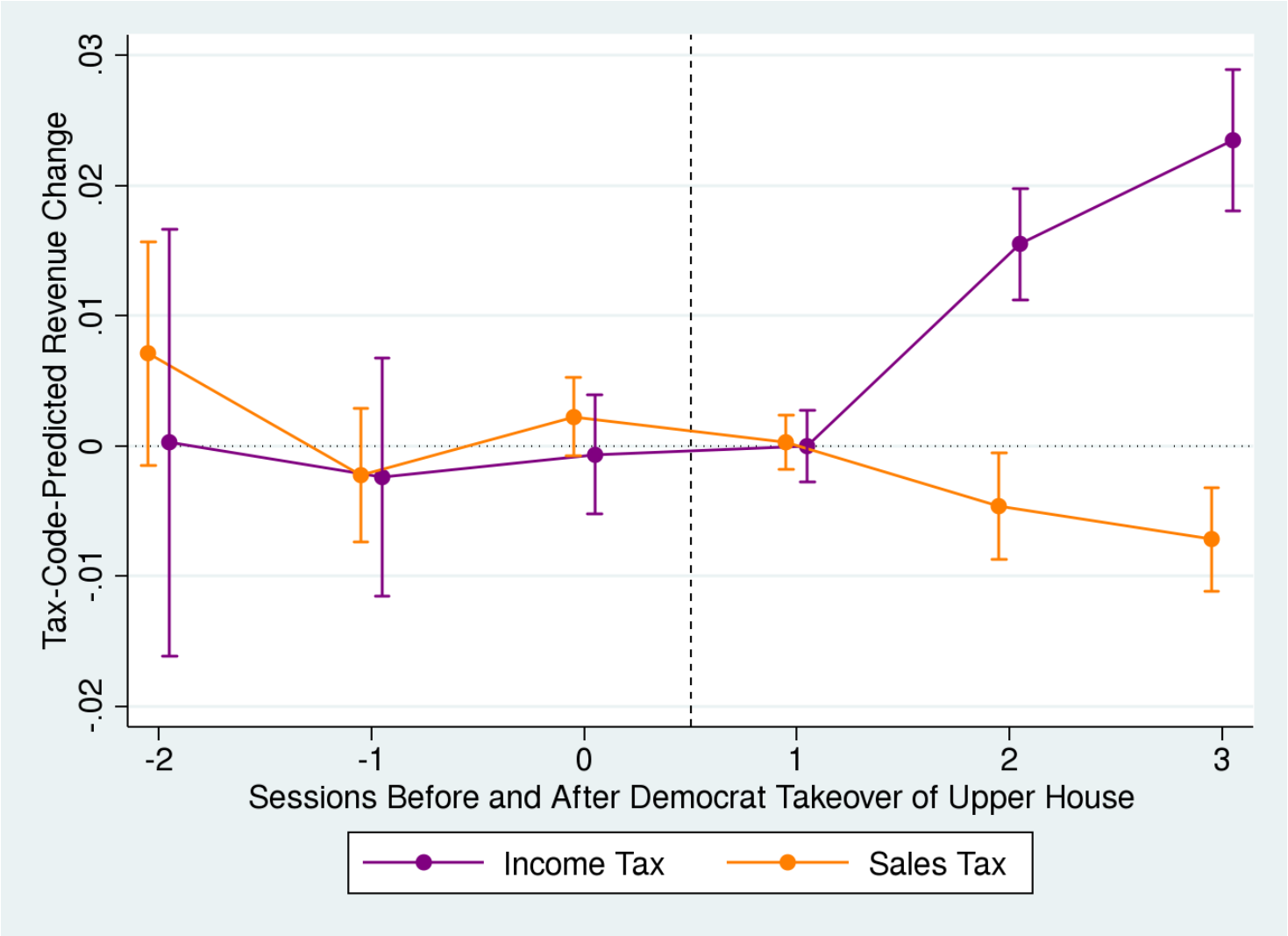
Sales Tax



Weak predictors filtered out; 80% training, 20% testing sample.

- Predicted change in revenue (vertical axis), plotted against true change in revenue (horizontal axis).
- Correlations between truth and prediction: 0.89 and 0.84.
- Source: Ash (2016)

Effect of Democrat Takeover on Tax Code Language



Event study graphs for change in text-predicted revenue before and after Democratic takeover of upper house of legislature. The vertical axis is the metric for state-predicted revenue $\tilde{g}$, as described in the text. The horizontal axis is years before and after a change in political control. Republican takeovers are also included, with the sign of the outcome variable reversed. Source: Ash (2016).