Deep Learning of Political Texts

Andrew Peterson

11.10.2017

Université de Genève

Introduction

Table of contents

Motivation

- Estimating ideological position from texts (speeches, bills, etc):
- Constrained, formal texts $\rightarrow$ difficult to identify position $\perp$ topic
- Researcher degrees of freedom in text analysis
- Estimating uncertainty

Syntax and Meaning

- “I will not raises taxes or allow imports.”
- “I will raise taxes or not allow imports.”

Consider two-word ‘speeches’

- ‘free healthcare’
- ‘free market’

Hypothetical Document Term Matrix

	free	externalities	heathcare	market	label
document 1	1	0	1	0	L
document 2	1	0	0	1	R
document 3	0	1	1	0	R
document 4	0	1	0	1	L

Possible Approach 1

- least squares

Least squares

$$y = \beta_1 * \text{free} + \beta_2 * \text{externalities} + \beta_3 * \text{healthcare} + \beta_4 * \text{market} + \epsilon$$

$\rightarrow$ random guess

Possible Approach 2: SVM

- map features into higher dimensional space

$$RBF(x, x_i) = exp(-\gamma||x - x_i||^2)$$

(kernel trick: don't actually have to generate mapping, just use distances)

RBF Transformed Document Term Matrix

	x1	x2	x3	x4
document 1	1.00	0.61	0.61	0.37
document 2	0.61	1.00	0.37	0.61
document 3	0.61	0.37	1.00	0.61
document 4	0.37	0.61	0.61	1.00

$\rightarrow [x1 \ldots x4] = [3.23, -3.23, -3.23, 3.23]$

Possible Approach 3: Deep neural networks

- So SVM works in this simple case.
- But: with some data better generalization, fewer parameters with a deep network. (Montufar et al. 2014; Ibarz et al. 2013).

Basics of Deep Neural Networks

Why Deep Learning?

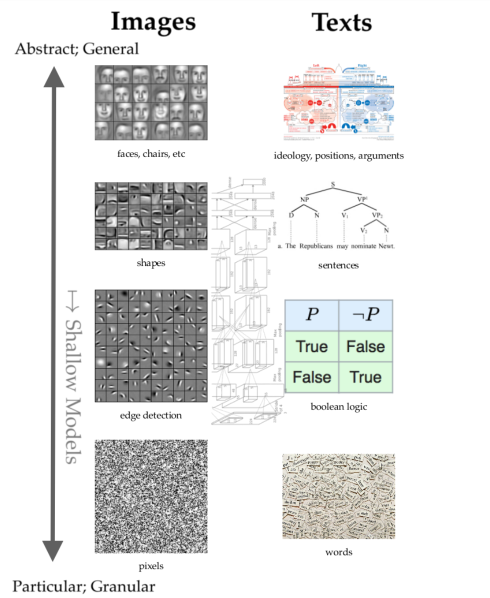
- Unifies feature identification and prediction into one step $\rightarrow$ general learning model

- multiple layers:
  $\rightarrow$ successively higher levels of feature abstraction

  $\rightarrow$ interactions between non-local features and logical relationships between them

  $\rightarrow$ translation invariance

Deep Neural Networks



Deep neural networks

- as function approximation: nested vector-valued functions $f(x) = f_2(f_1(x))$, which introduce non-linearities.

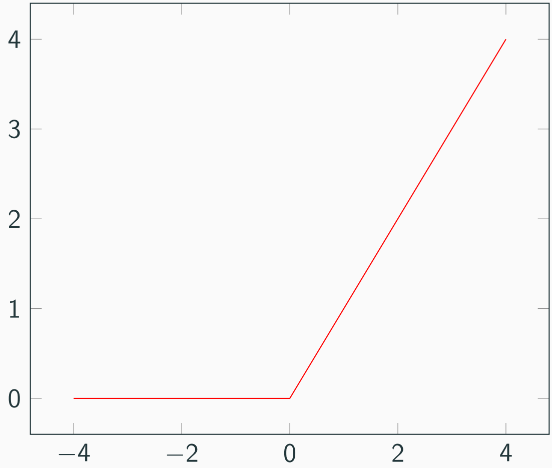
Simple neural network

$$\text{layer 1 output: } Z = max\{0, (X \cdot M + b_1)\}$$

$$\text{layer 2 output} = \text{sigmoid}(Z \cdot v + b_2)$$

where $\text{sigmoid}(x) = \frac{1}{1+exp(-x)}$.

Nonlinear activation -Rectified linear Unit function



$$\text{layer 1 output} = ReLU(X \cdot M + b_1)$$

```
import numpy as np
from keras.models import Sequential
from keras.layers.core import Dense

# enter the matrix from Table 1
document_term_matrix = np.array(
[[1,0,1,0],
[1,0,0,1],
[0,1,1,0],
[0,1,0,1]],
 "float32")

y = np.array([[1],[0],[0],[1]], "float32")

model = Sequential()
model.add(Dense(8, input_dim=4, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='mean_squared_error', optimizer='adam')

model.fit(document_term_matrix, y, epochs=200)

print model.predict(document_term_matrix).round()
```

DNN for Texts

Alternative to Bag-of-words: Incorporate order

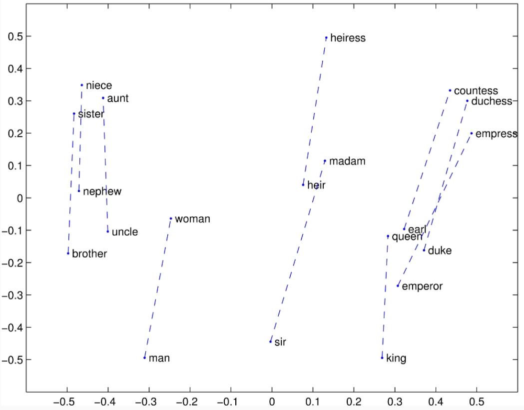
- Can preserving sentence order help identify *position* on a topic?

Problem of High Dimensionality

- # vocab * sentence length (e.g. 20,000 * 25 = 500,000)
- New approaches: vector word embeddings

Vector word embeddings

- Represent each word by a 100-dimensional real-valued vector
- train vectors to e.g. predict neighboring words



Source: Pennington, et al. (2014)

Problem: Bag-of-words all equidistant

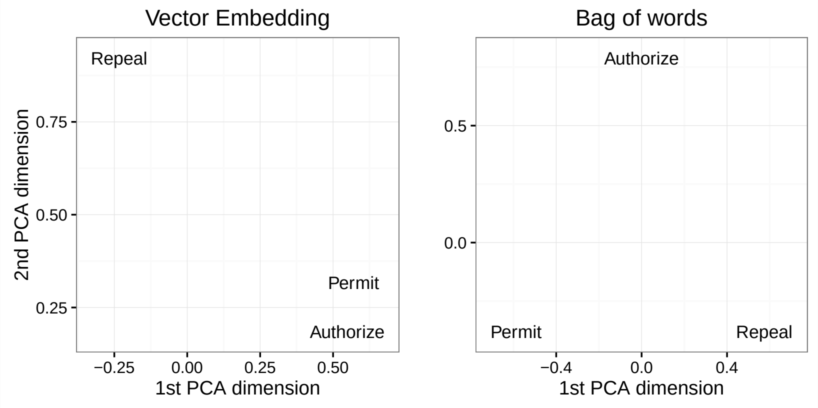
- Document term matrix:

$$\begin{aligned} \text{authorize} &= [1, 0, 0, 0, \ldots, 0] \\ \text{permit} &= [0, 1, 0, 0, \ldots, 0] \\ \text{repeal} &= [0, 0, 1, 0, \ldots, 0] \end{aligned}$$

- vector word embeddings:

$$\begin{aligned} \text{authorize} &= [-0.5, 0.1, -0.3, \ldots] \\ \text{permit} &= [-0.4, 0.05, -0.2, \ldots] \\ \text{repeal} &= [0.2, 0.1, 0.7, \ldots] \end{aligned}$$

First two dimensions of PCA



Simulation with Predicate Logic

Simulated Speeches

- 80 word speeches.
- meaning of words not additive, atomistic; 2-place predicate logic
- three types of words: partisan (1,000), function (4), and noise (18,996)

Two-place functions

$$
\begin{aligned}
f_1(x, y) &= (x + y) \\
f_2(x, y) &= (-x - y) \\
f_3(x, y) &= (x - y) \\
f_4(x, y) &= (-x + y)
\end{aligned}
$$

Simulated Parliament

- Draw speaker ideal points, leadership
- For each speech: draw speaker, ideal point, function, partisan words, random words

Models

- SVR
- 2-layer neural network
- CNN-LSTM

CNN-LSTM (details)

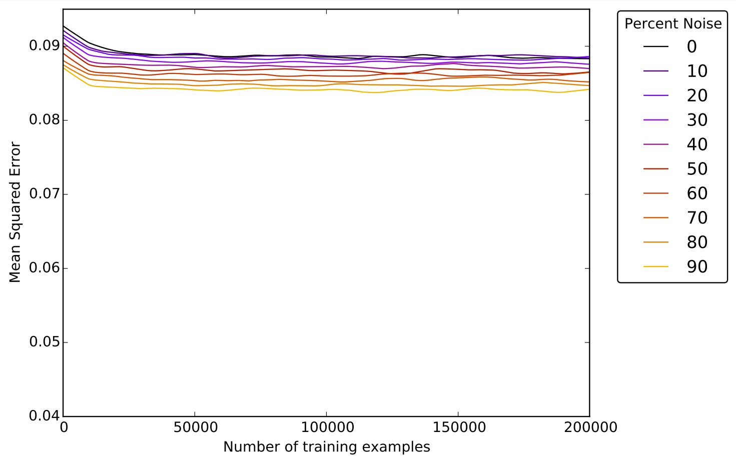
- 128-dim embeddings, CNN, max pooling, LSTM, dense
- loss: mean squared error
- SGD, adam (adaptive learning rate)
- "Concrete dropout" (Gal, et al 2017)

Results

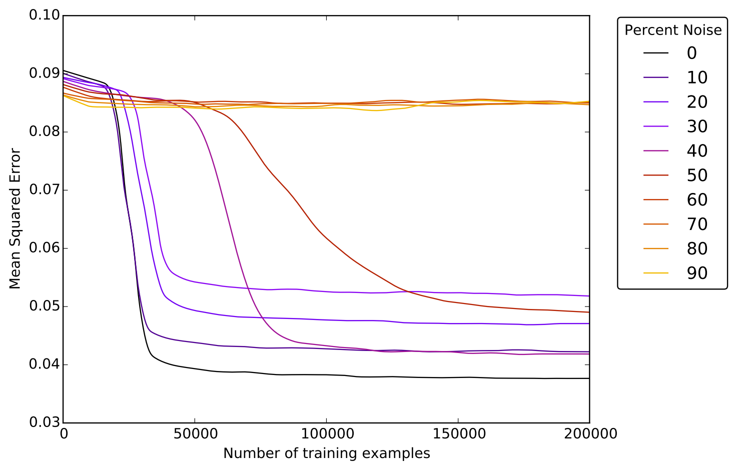
Mean squared error by method, no added noise

	(guess mean value)	0.099
1	SGD regression	0.093
2	SVR	0.091
3	2-layer NN	0.088
4	CNN	0.087
5	CNN-LSTM	0.037

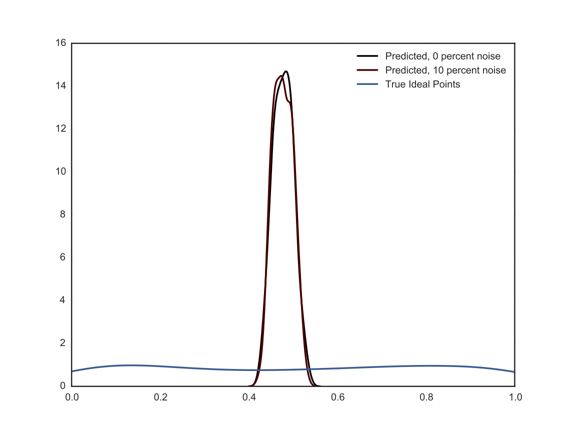
Mean Squared Error, 2-layer NN



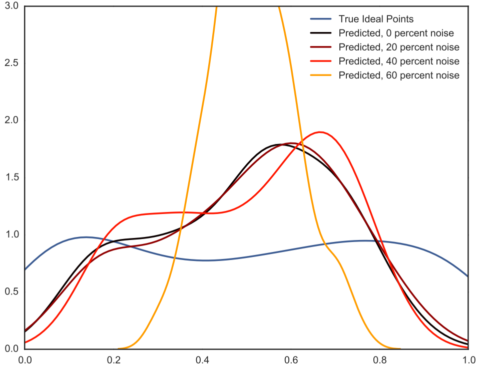
Mean Squared Error, CNN-LSTM



Estimated and True Ideal Points, SGD



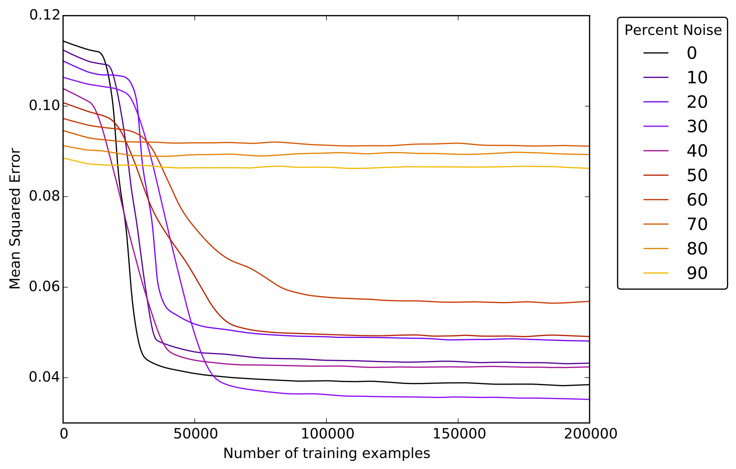
Estimated and True Ideal Points, CNN-LSTM for Different Levels of Noise



Conclusion

- Applied to UK Parliamentary speeches, US legislation
- Estimating uncertainty

Mean Squared Error, CNN-LSTM, more polarized data



Questions?

Artificial Bill Text, Generated Character-by-Character

(Recurrent neural network using 6MB of text)

Assistance to Charren Contract Development Procedures.–

(1) In general.–Section 2963(b)(3) of title 5, United States Code, is amended by adding at the end the following new paragraph:

"(7) In addition to any other payment on a given trauma center thereof, or the failure of a national financial assistance under this section for discretionary budget authority to settle the reasonable contributions requested under section 1980.

"(b) Grants to Minority Penalty.–The uninsured technical or environmental [...]