DRAFT

Unsupervised Text Classification With Chinese Social Media

An Extension of King, Pan, and Roberts (2017)

Abstract

<150 words

I. Introduction

- Background on the 50c party. State of knowledge before KPR (2017) and since.
- Other quantitative approaches to studying the 50c party and Chinese internet "opinion guidance."
- Summarize King, Pan, and Roberts (2017), specifically data source and results.
- Try to improve figures if I can replicate the original ReadMe results (but probably don't have the human-coded posts).
- Describe impact/significance of the paper.

II. Extension

- Motivation/rationale for why the reassessment is warranted.
- Categories are informed by existing knowledge but still somewhat arbitrary…
- Define research question.
- How unsupervised text classification works (general approach) vis-à-vis ReadMe.
- Why I picked certain methodologies.

IDEAS * Automate + boost ReadMe (run it many times with different classifications?) * Test my results with the out-of-sample batch. (knownWeibos_zg)

III. Results and Discussion

- What the findings were.
- Implications of findings.
- Directions for future work.

Acknowledgements

Technical Appendix

- Basically, explaining my code. Why I made certain decisions.

The approach I took is detailed in Stanford (https://web.stanford.edu/~gentzkow/research/text-as-data.pdf, 5). In the document term matrix (DTM), I define $\mathbb{D}$ as the set of individual

posts $\{\mathbb{D}_i\}$ with row c_i the numerical vector that represents the presence or weight of a particular language token j. I remove stop words (from a list of 750+) and punctuation.

I then reduce the dimensionality of the DTM from around $i \approx 22,000$ to $i \approx 2,000$ with two methods: first by per-document word count, and second by term-frequency-inverse-document-frequency (tf-idf), which excludes both common and rare words.

Term frequency is

$$tf_{i,j} = \frac{n_{i,j}}{\sum n_{i,j}}$$

where $n_{i,j}$ is the number of occurences of token j in post i and $\sum_k n_{i,j}$ is the total number of occurences of the token through all of the documents.

Inverse data frequency is

$$idf = \log\left(\frac{N}{df_i}\right)$$

where df_i is the number of documents containing i out of all of the N documents, so rare words have a high idf score.

Tf-idf is the product of the two terms, $tf \times idf$.

I did not attempt to "stem" the words further because Chinese words do not require stemming (some possible exceptions are nouns and verbs that end in "- " or "- ," respectively).

References