Part II: Analyzing Text as Data

Solomon Messing

Department of Communication, Statistics
Stanford Social Science Data and Software (SSDS)

September 24, 2012

What can we do with all this text?

- Interpret meaning
  - Edge: humans
  - But natural language understanding is getting better (IBM Watson)
- Classify, detect, compare
  - Edge: machines
  - Why?
- Combine with temporal, behavioral, locational, other data sets

For more see Grimmer's Text as Data Course

http://stanford.edu/~jgrimmer/tc1.pdf

Text is complicated!

"I believe that I interpret the will of the Congress and of the people when I assert that we will not only defend ourselves to the uttermost, but will make it very certain that this form of treachery shall never again endanger us."
-FDR

- How many features do we need to capture everything that is happening?
- Can we define rules that capture what each word means in context?
- How is a machine going to know *who* has committed this treachery (i.e., the Japanese)?

Text is complicated!

Other problems:

- Verbal Irony - “as pleasant and relaxed as a coiled rattlesnake” (quote from Vonnegut)
- Subtle negation - “They have not succeeded, and will never succeed, in breaking the will of this valiant people” (Janyce Wiebe)
- Order Dependence - “Peace, no more war” v “War, no more peace” (Arthur Spirling)

Features

- Let's scope the problem: delineate a unit of text (sentence, paragraph, or "document")
- Define a feature that records the presence or absence of something (word, words, grammar, entity, etc.).
- Still $\rightsquigarrow$ high-dimensionality (too many variables)

Features: lowering dimensionality

We can collapse variables that are basically the same and remove others. Some meaning is lost, but the problem becomes more tractable.

- 'congress' & 'Congress' $\rightsquigarrow$ remove capitalization, punctuation.
- Do we need words like 'a,' 'the,' 'and'? $\rightsquigarrow$ remove stop-words. NB you'll often get better performance in short texts (e.g., Tweets) without removing these...
- 'sleep' & 'sleeping' - $\rightsquigarrow$ stem to remove endings.
- Discard non-discriminating words (now stems).
  - Rarely occurring stems, $< 1\%$
  - Commonly occurring stems, $> 99\%$ (or $> 99.9\%$)
- Or re-weight words to emphasize discriminating words (e.g., TF-IDF weighting).

Features: Bag of words assumption

Discard word order.

"Now we are engaged in a great civil war, testing whether that nation, or any nation"
-Lincoln[1]

[1] This example is from Grimmer's Text as Data course: http://stanford.edu/~jgrimmer/tc2.pdf

Features: Bag of words assumption

Unigram counts of words:

Unigram	Count
a	1
any	1
are	1
civil	1
engaged	1
great	1
in	1
nation	2
now	1
or	1
testing	1
that	1
war	1
we	1
whether	1

Features: Bag of words assumption

Bigram counts of words:

bigram	count
now we	1
we are	1
are engaged	1
engaged in	1
in a	1
a great	1
great civil	1
civil war	1
war testing	1
testing whether	1
whether that	1
that nation	1
nation or	1
or any	1
any nation	1

Features: Bag of words assumption

We've stripped out most of what makes this speech great, defiling arguably the nation's most sacred text. Can doing this possibly predict anything we care about? Yes, at least sometimes.

- Words alone can tell us about the *topic* of text, if not the sentiment.
- Subtle negation is less common that you'd think—and substantial performance increases with simple rules, e.g., simply transforming "not good" to "not_good" (Pang & Lee).
- Order dependence is rare enough that it doesn't always kill us.

Features: Parts of speech

OK, we've got word counts. But what about polysemy: "this situation is grave" versus "this situation is going to put me in an early grave?"

- We can use a probabilistic POS tagger to tell us how a word is being used.
- Uses Viterbi Algorithm (HMM or dynamic programming), see http://en.wikipedia.org/wiki/Part-of-speech_tagging
- Does this fully solve our problem? What does the second expression really mean?

Features: Entites

We might also want to know something specific about proper nouns mentioned in text—does sentiment toward Egypt differ by Democrats and Republicans?

- We can use regex to define our own entity features and/or use wordlists. Here's one example implemented in R.
- Or we can use probabilistic named entity recognition (NER)—not implemented in R (yet), but see Stanford's named entity tagger, the same at Illinois, and check out Yahoo's API which has very complex entity tagger.
- Then use in other analyses (e.g., Tweets that mentioned Eygpt to Republican representatives tended to be about about violence while those to Democratic representatives tended to be about good governance).

OK, now what?

We have text- and possibly other features that we want to use to learn something general about the content.

- Supervised learning - train a model to classify documents based on labels.
- How do we get these labels?
  - Human coder content analysis. If the task is extremely simple, you may wish to use Mechanical Turk, if not, the Coding Analysis Toolkit has a nice interface.
  - Or use a label that is already available (party of speaker, hashtag, emoticon, etc.).
- Unsupervised learning - cluster this data to create new labels, with minimal assumptions.

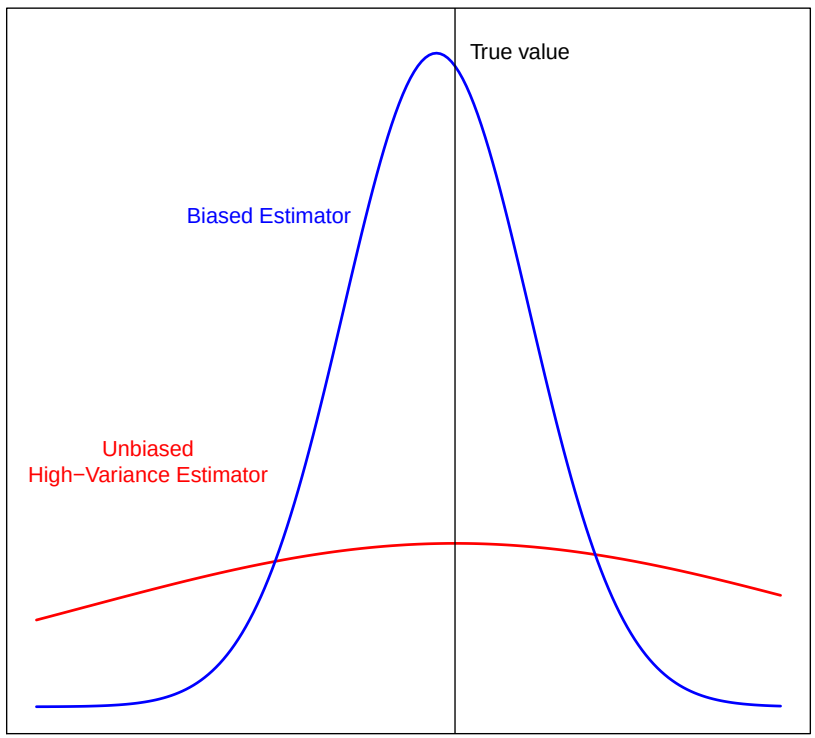
Supervised Learning

- We have a matrix of textual units x features (often called a document-term matrix). So we can use our friend, linear regression, to model it right?
- Well no, the curse of dimensionality gets us, despite the fact that we reduced it above.
- What is a dimension?
- What is this curse?
- Sampling density $\propto N^{\frac{1}{p}} \rightsquigarrow$ variance of our predictions will increase with each new feature (while bias will decrease).

Supervised Learning

How do we deal with the curse of dimensionality?

- Increase bias in favor of lowering variance. This isn't a 1:1 relationship; we can often do much better with a little bit of bias.



Supervised Learning

How do we deal with the curse of dimensionality?

- A little bias can go a long way.
- Ridge to shrink large values of one coefficient toward another with which it's correlated.
- LASSO to zero out βs that perhaps shouldn't be in the model.
- Elastic Net to combine these approaches (see `glmnet`).
- Many many other approaches (Naïve Bayes, SVM, Decision Trees, Boosting, Bagging, etc.), most also increase bias in favor of lowering variance.
- Cross-validation necessary when evaluating results to avoid selecting a model that is overfit to the data.

Supervised Learning: Cross-validation

- Fit a model to a *training set* (bigger subset of labeled data).
- Evaluate model performance a held out *test set* (smaller subset of labeled data) using MSE, Accuracy, Precision, Recall, F, AUC or something else.
- Choose different subsets and repeat.
- Compute mean and 95% CI for test statistic(s).
- Repeat for different tuning parameters (e.g., λ)

Supervised Learning: Classification

- We now have a model of the alleged relationship between our features and our outcome
- Apply the model to new text data, and get your predictions. Time to go home right?
- No. Validate, validate, validate, even if only with (randomly selected) anecdotal examinations of machine-labeled/machine-scored documents.

Lab 3

Lab 3: Analyzing text as data.

Other Resources

- This was only a taste.
- Justin Grimmer's Text as Data course, forthcoming Political Analysis paper on Text as Data and PNAS paper on computer-assisted clustering. Excellent treatment of these and other methods for analyzing text, especially unsupervised techniques which are not covered here.
- CS 124 for more on Natural Language Processing.
- Hastie, Tibshirani, & Friedman's The Elements of Statistical Learning, an excellent book on ML which they have generously made available online without cost.
- The Machine Learning for Hackers Codebase for more examples of text analysis in R.