Text Analysis with R, part 3

Michele Claibourn, UVA Library's StatLab

November 13, 2018

Roadmap for today

- topic modeling - latent dirichlet allocation
- lexicon-based classification - sentiment
- model-based classification - naive Bayes,

Still using Quanteda, along with friends.

Today's Text

1. Comments submitted to Ours to Shape
   - scraped with `acquire_comments.R` (in materials)
2. UVA course descriptions for last several semesters
   - downloaded from Lou's list

R Script Supplement | Topic Model, LDA

To aid in automatic discovery of thematic content: We observe documents and words; topics are part of latent (hidden) structure we wish to infer. Given a number of topics:

- Across a fixed vocabulary, each topic is a distribution over terms
- Each document is a distribution over topics
- To generate a document, randomly choose a distribution over topics. For each word in the document
    - Probabilistically draw one of the k topics from the distribution over topics
    - Given the topic, draw a word from the distribution over terms
    - Rinse and repeat

The model estimation “reverses” this stylized stochastic process to infer estimates of the originating topic and term distributions.

R Script Supplement | Classification

Goal is to place documents into a pre-defined categories.
Classes/categories could encompass

- Topics, e.g., policy areas for legislation
- Authors, e.g., stylometry
- Spam or other filters
- Sentiment, opinion, tone, e.g., negative/positive, or specific emotional expression
- Any latent (hidden) construct defined by language...

Multiple approaches, same task

- Dictionaries: pre-identified words that associate with classes are counted/weighted
- Supervised classification: statistical models identify separating words

R Script Supplement | Lexicon-Based Classification

Many “off-the-shelf” dictionaries available, for example

- Linguisting Inquiry Word Count, LIWC (Pennebaker et al 2001), measures 82 language dimensions
- General Inquirer Database (Stone et al 1966), 182 categories
- Affective Norms for English Words, ANEW (Bradley and Lang 1999), three semantic differentials (good-bad, active-passive, strong-weak)
- Lexicoder Sentiment Dictionary, LSD (Young and Soroka 20120), negative to positive tone
- Moral Foundations (Haidt et al 2009), virtue and vice words for harm, fairness, ingroup, authority, and purity

To work, the dictionary weights must align with how words are used in the context under study!

R Script Supplement | Lexicon-Based Classification

Given a vector of word counts $X_i = (X_{i1}, X_{i2}, \ldots, X_{ik})$ and weights attached to words $\theta = (\theta_1, \theta_2, \ldots, \theta_k)$ each document, i, score is calculated

$$d_i = \sum_{i=1}^{n} \theta_k X_{ik}$$

If documents are of varying length, some normalization (e.g., divide by word count).

- Dictionary methods are context invariant; same word weights regardless of texts
- Easy to use, but should be coupled with validation – face validity of words, examination of classified documents, etc.
- Modifiable areal unit problems and ecological fallacies

R Script Supplement | Model-Based Classification

Human coders classify a subset, N_{train} documents, into predetermined categories (or text is conveniently pre-coded).

- Need clear categories, simple coding rules, and trained coders!
- Multiple coders for (at least some) documents to test inter-coder reliability (e.g., Krippendorf's α, Cohen's κ).
- Produce a labeled set for training, a labeled set for validation. How many?

Labeled documents are used to train a model, optimize with respect to θ to "learn" the weights. Model is validated against hand-labeled test data N_{test} by comparing predicted fit to pre-labeled categories.

R Script Supplement | Naive Bayes Classifier

A simple application of Bayes' rule, surprisingly useful!

For each document i , we want to infer the most likely category, C_k, based on features of the document x_i

$$C_{Max} = argmax_k p(C_k|x_i)$$

Use Bayes' rule to estimate $p(C_k|x_i)$

$$p(C_k|x_i) = \frac{p(C_k, x_i)}{p(x_i)} = \frac{p(C_k)p(x_i|C_k)}{p(x_i)}$$

Estimate $p(C_k) = \frac{docs\ \ in\ \ k}{docs\ \ in\ \ N_{train}}$. Estimating $p(x_i|C_k)$ is complicated . . . unless we make the naive assumption that features, x_i, are independent.

Given independence $p(x_i|C_k) = \prod_{i=1}^{N} p(x_i|C_K)$

R Script Supplement | Confusion Matrix and Related Metrics

	Predicted	
Actual	Positive	Negative
Positive	True Pos	False Neg
Negative	False Pos	True Neg

$$\text{Accuracy} = \frac{TruePos+TrueNeg}{TruePos+TrueNeg+FalsePos+FalseNeg}$$

$$\text{Precision} = \frac{TruePos}{TruePos+FalsePos}$$

$$\text{Recall} = \frac{TruePos}{TruePos+FalsNeg}$$

$$\text{F1} = 2 \times \frac{Prescision \times Recall}{Precision+Recall}$$

R Script Supplement | Logit with Lasso Regularization

Logit

- Predict $p(y = 1)$; because probabilites are bounded, use a cumulative probability distribution - an S-shaped curve
- Select a threshold to map probabilities into binary outcome, e.g., $p > 0.5 = 1$ and $p < 0.5 = 0$

Regularization

- Lots of features, $p >> n$ problem; and features are highly correlated; leads to more variable estimates and overfitting
- Regularization constrains the coefficient estimates, shrinks them towards zero

The Lasso

- Uses all predictors, shrinks less important ones to 0
- Adds a tuning paramter, λ, as penalty to model complexity; when $\lambda = 0$ Lasso produces least-squares/MLE fit; as $\lambda \to 1$ all coefficients approach zero

Concluding Thoughts

1. All quantitative models of language are wrong, but some are useful
2. Validate, validate, validate