Topic Modeling

Things you should read before publishing anything:

- Those three Coursera courses you have for a week
- The relevant Grimmer chapters

Introduction

This post will cover topic modeling, starting with basic word co-occurrence clustering techniques not usually considered topic modeling and ending with complex models

What are the uses of topic modeling? - Get some of these from Silge and Robinson. Others from Blei. - You can classify documents. This is maybe the best-case scenario. You fit the model on unlabeled documents, you examine the betas and see really clearly that topic 1 is [thing], topic 2 is [other thing], etc. Then you have all your documents classified, tagged, as it were. You can then use that to filter (as I did to find Ezra's political episodes)

What are the data requirements? Sure, more is better, but what else can you say?

Word Co-Occurrence

Preprocessing

Bind entities with "_" (e.g., "Donald_Trump" instead of "Donald Trump").

Interpreting Model Parameters

Betas are word (token)-topic probabilities. This is the probability that this word will be generated by this topic. If you don't remove stopwords, the words with the highest betas will be stopwords. If you do remove stopwords, the words most like stopwords will usually be highest. This is why a good preprocessing step might be to remove high-frequency words, especially those that have a low idf, in that corpus (see preprocessing bit)

Klein Case Study

Much like philosophers, we first have to make some ontological decisions. First, given a corpus, what's a document? Second, how many topics were employed in generated the text of this corpus?

What's a document? The extremes are pretty easy: You have individual lines at one end, then you've got entire episodes at the other end. Another, maybe more realistic approach would be to find natural breaks within episodes.

But you might wonder how important these considerations are given that the LDA algorithm lets documents arise from *mixes* of topics. So what would be the harm in allowing for documents to be the most expansive possible unit? I'll just mention one tradeoff. Because the algorithm considers documents as unordered bags of words, it doesn't pay attention to the fact that words closer to each other are more likely to be generated by the same topic.

When at how many k does a political topic emerge? When does it bi-, tri-, quadrufurcate, etc.? - And what are the different contents of these?

Show your code for fitting many different k's at once and then getting the per-document median GINI (mean)

Forthcoming: Pre-trained embeddings then clustering

Assorted Notes

When you have multiple keywords per entry on a news site, that's kind of like saying multiple topics! The real world matches LDA's DGP there.

What are the assumptions of topic modeling? Remember that it's an unsupervised learning, so when talking about assumptions and uses, it's probably going to inherit some of those.